A FRAMEWORK FOR SIMULATION-BASED INTEGRATED DESIGN OF MULTISCALE PRODUCTS AND DESIGN PROCESSES

A Thesis
Presented to
the Academic Faculty

By

Jitesh H. Panchal

In Partial Fulfillment
of the Requirements for The Degree of
Doctor of Philosophy in Mechanical Engineering

Georgia Institute of Technology
December 2005

A FRAMEWORK FOR SIMULATION-BASED INTEGRATED DESIGN OF MULTISCALE PRODUCTS AND DESIGN PROCESSES

Approved by:

Dr. Farrokh Mistree, Advisor
GWW School of Mechanical Engineering
Georgia Institute of Technology

Dr. Christiaan J.J. Paredis
GWW School of Mechanical Engineering
Georgia Institute of Technology

Dr. Janet K. Allen
GWW School of Mechanical Engineering
Georgia Institute of Technology, Savannah

Dr. David L. McDowell
GWW School of Mechanical Engineering
Georgia Institute of Technology

Dr. David Rosen
GWW School of Mechanical Engineering
Georgia Institute of Technology

Dr. Chuck Eastman
Colleges of Architecture
Georgia Institute of Technology

Dr. Kwok Tsui
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: Nov 22, 2005

To my Parents
(Mrs. Indu Panchal and Mr. Harshad C. Panchal)

for all their love and sacrifice for me

ACKNOWLEDGEMENTS

This dissertation would not be possible without the immense support and encouragement from various people.

First of all, I would like to thank my Ph.D. advisors Farrokh Mistree and Chris Paredis for their continuous support over the past few years. I am very fortunate to have both Farrokh and Chris as my advisors. Their different personalities and perspectives have not only provided me a broader understanding of various research topics, but also provided me reasons to think critically.

It is difficult to overstate the gratitude of my mentor, Farrokh, who has been like a father for me. His enthusiasm and love for students is truly remarkable. He has a unique ability to help students realize their full potential. Farrokh has helped me by setting high goals and providing support thoughout the journey to achieve those goals. His visionary ideas and invaluable insights have always challenged me to go beyond the 'low hanging fruits'. I appreciate his guidance balanced by the freedom to express myself. I believe that he is the best mentor a student can have.

Chris Paredis has been more than a research advisor. He has been like a friend to me. He has guided me throughout the process of my PhD dissertation research. I am amazed by his willingness to talk to the students and think along with them. He has encouraged me to be strongly grounded in fundamentals and to ask fundamental questions.

I am indebted to Janet K. Allen for helping me with the funding and making me a part of the Materials Design project. I am grateful to her for sharing her knowledge on Uncertainty and Robust Design, which has been a significant help for me. I would like to extend my warmest thanks to David McDowell for his advice on my research. His vision for design of multiscale materials is a key to the conceptualization of this dissertation. My discussions with David provided me confidence about the impact of this research in new design applications. I would also like to thank my dissertation committee members David Rosen, Chuck Eastman, and Kwok Tsui for their insightful comments. Their comments have helped me in focusing my research significantly.

I would like to thank HaeJin Choi, Marco Fernandez, with whom I have worked closely during my PhD research. HaeJin and Marco have been great colleagues. They have always been available for helping their colleagues in a variety of ways ranging from writing papers and sharing research ideas, to helping with computer code. I would like to thank Nathan Rolander for getting me started on the datacenter example, and teaching me various software applications such as Fluent and FemLAB; Carolyn Conner Seepersad for helping me with the Linear Cellular Alloy example and providing me the code for thermal and structural analyses; Jim Shepherd for taking the lead on integrating simulation applications in the materials design project; and Ryan Austin for providing the Shock Simulation code.

I thank all the members of the Systems Realization Laboratory including Chris Williams, Matt Chamberlain, Rakesh Kulkarni, Greg Mocko, Yao Lin, Andrew Schnell, Stephanie Campbell, Hannah Muchnick, and others for contributing to a research environment that has proven immensely rewarding. I want to thank my friends Manas Bajaj, Sanjeev Dwivedi, Arnab Choudhury, Harpreet Singh, and Abhina Saxena for providing an environment, where I never felt away from home.

Above all, I would like to thank my family members for all the love they have given me. Without them, I would be nothing. I cannot thank my parents enough for all the sacrifice they have made for me. I also want to thank the new member of our family, Papori, who has been a tremendous source of support for me. I thank her for her love, encouragement and her confidence in me over the last six years.

The work presented in this dissertation has been supported by NSF Grants DMI-0100123 and SES-0085189, AFOSR grant F49620-03-1-0348, AFOSR-MURI grant 1606U81 and ONR grant N00014-01-1-0267. I am grateful for the financial support from these organizations without which this work would not be possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Problem: The complexity in multiscale systems design is significantly greater than in conventional systems because in addition to interactions between components, couplings between physical phenomena and scales are also important. This complexity amplifies two design challenges: a) complexity of coupled simulation models prohibits design space exploration, and b) unavailability of complete simulation models that capture all the interactions. Hence, the challenge in design of multiscale systems lies in managing this complexity and utilizing the available simulation models and information in an efficient manner to support effective decision-making.

Approach: In order to address this challenge, our primary hypothesis is that the information and computational resources can be utilized in an efficient manner by designing design-processes (meta-design) along with the products. The primary hypothesis is embodied in this dissertation as a framework for integrated design of products and design processes. The framework consists of three components – 1) a Robust Multiscale Design Exploration Method (RMS-DEM), 2) information-economics based metrics and methods for simplification of complex design processes and refinement of simulation models, and 3) an information modeling strategy for implementation of the theoretical framework into a computational environment.

The first research question is related to the configuration of design processes for effective design decision making. The hypotheses used to answer this research question are: *a)* design processes can be refined systematically along with the products, and *b)* design processes can also be designed as modular systems composed of repeating

building blocks. The second research question is related to systematic simplification of design processes and the extent of refinement of simulation models. The hypothesis used to answer this research question is that information economics based metrics can be used to quantify the impact of design process related decisions. The third research question is related to the modeling of design information to support design of design processes. The hypothesis used to answer this question is that separation of product, decision problem, and design process related information enables design of design processes in computational frameworks.

Validation: The framework is validated using the validation-square approach that consists of theoretical and empirical validation. Empirical validation of the framework is carried out using various examples including: pressure vessel design, datacenter cooling system design, linear cellular alloy design, and multifunctional energetic structural materials design.

Contributions: The contributions from this dissertation are categorized in three research domains: a) multiscale design methodology, b) materials design, and c) computer-based support for collaborative, simulation-based multiscale design. In the domain of design methodology, new methods and metrics are developed for integrating the design of products and design processes. The methods and metrics are applied in the field of materials design to develop design-processes and specifications for Multifunctional Energetic Structural Materials. In the domain of computer-based support for design, an information modeling strategy is developed to provide computational support for meta-design. Although the framework is developed in the context of multiscale systems it is equally applicable to design of any other complex system.

Chapter 1 Integrated Design of Products and Design Processes for Multiscale Systems

The principal goal in this dissertation is to establish a framework for integrated design of multiscale products and design processes, to facilitate effective and efficient utilization of information and computational resources in preliminary design of multiscale systems with potential applications to other complex systems.

The motivation for this research is the need for systematic simulation-based design methods suitable for designing systems by appropriate consideration of phenomena at various scales. As pointed out in Section 1.1, multiscale modeling is an evolving multidisciplinary field that integrates constructs from mathematics, computational science, and specific engineering domains (such as material science, bio science, environmental science, etc.). The primary research objective in multiscale modeling is to employ simulation models at different scales (length, time, energy, etc.) to gain a complete understanding of phenomena that could not be modeled otherwise. In this dissertation, the multiscale modeling efforts are taken a step further into *multiscale design*, where these models are utilized for the design of complex engineered systems and the associated decision making. A characteristic of simulation-based design of multiscale systems is that there are a number of simulation models that generate different fidelities of information about various aspects of the system. The complexity in design of multiscale systems arises from the coupling of phenomena at different scales leading to the overall behavior of the system. Ideally, from an accuracy perspective, a system representation that incorporates all the interactions between all scales would be preferable. However, there are two barriers in using completely coupled system models at all the scales – *a)* these coupled models are computationally very expensive and hence, unfit for designing reasonably sized engineering systems, and *b)* such completely coupled models that capture all the interactions between systems are generally not available. Nevertheless, from a decision making perspective, it is not a very big barrier because most of the couplings have only a small effect on the design decisions and only a few couplings are important to generate models that are *good enough* for decision making. This is especially true in the preliminary design stages. Hence, the challenge in simulation-based multiscale, multifunctional design is to systematically account for interactions that support effective decision making. The primary hypothesis to address this challenge is that "*simulation-based design of multiscale, multifunctional systems can be carried out by decision-based integrated design of products along with their design processes*".

Designing simulation-based design processes from a decision-based perspective involves answering questions such as:

a) What is the sequence in which product decisions should be made?

b) Which models should be used for making decisions?

c) Which interactions are important for making decisions and which interactions can be ignored?

d) What level of accuracy in models is appropriate for decision making?

e) How can existing knowledge be reused in new design scenarios?

The starting assumption in this dissertation is that by answering these questions and appropriately configuring the design processes (along with design of products), we can also improve the efficiency and effectiveness with which these products are designed. In order to answer the questions related to designing design processes, a multiscale design framework is developed in this dissertation. The framework consists of three components: *a)* a Robust Multiscale Design Exploration Method (RMS-DEM) that consists of three steps – meta-design, design process execution, and refinement, *b)* metrics and methods for simplification of complex design process and simulation model refinement using information economics and robustness, and *c)* an information modeling strategy for simulation-based design information to support design process exploration and information reuse. Although the framework developed in this dissertation is important for multiscale systems, we believe that it is applicable to any complex system design.

The scope of applicability of constructs developed in this dissertation is *simulation-based preliminary design*. In the preliminary design stage, detailed simulation codes are available to predict computationally the behavior of systems. It is recognized that these simulation codes may have uncertainty associated with them and can be refined further. Preliminary design is carried out between the conceptual design phase, that is common to most design methods such as (Pahl and Beitz 1996), where the concept is not known and detailed simulation models are unavailable and the detailed design phase where the finalized design process is executed to generate manufacturable product description.

The multiscale design motivations are discussed in Section 1.1. The challenges with multiscale systems are separated into multiscale modeling challenges and multiscale design challenges. These challenges are abstracted into a comprehensive set of requirements for a multiscale design framework and a primary research question is framed. From these requirements, a subset of requirements that will be addressed in this dissertation is selected in Section 1.1.5. Three sub-research questions are formulated from the requirements in the Section 1.1.5. A high level overview of the strategy and hypotheses for answering each of the three research questions is presented in Section 1.2. Finally, a strategy for verification and validation is presented along with the outline of the dissertation in Section 1.3.

1.1 Multi-scale Systems – An Emerging Challenge for Engineering Systems Design

The excitement in the simulation-based design community over the past 2-3 years, as evident from the recent NSF workshop on Simulation-based Engineering Science (SBES) (SBES Workshop Report 2004), is attributed to the availability of independently developed simulation models at multiple scales of length and time. SBES represents an interface between diverse disciplines. SBES represents an interface between diverse disciplines, and has the following features – *a)* focus on using computer-based simulations for predicting system behavior, *b)* utilization of multiscale, multi-physics models, *c)* an engineering systems approach where multiple aspects of the problem are considered together, *d)* utilization of experience in engineering and applied mathematics, and *e)* a design/problem solving approach.

Following the NSF workshop on SBES, US Department of Energy sponsored three workshops on multiscale mathematics (Dolbow, Khaleel et al. 2004) to identify research and funding opportunities in multiscale modeling. During these workshops, a number of issues related to multiscale modeling were identified. From a multiscale modeling standpoint, the primary challenge is to *integrate information* generated by different simulation models in a consistent manner so that the overall system behavior can be predicted from the individual constituent models. During the two workshops, various application domains that would benefit from multiscale modeling were identified. These include environmental sciences, geosciences, climate, material science, combustion, biosciences, power grids and information networks, development of biomimetic sensors and devices, homeland security, etc. Since it is possible to predict the behavior of systems at multiple scales, the natural next step is to use these models for ***designing systems at multiple scales***. In contrast to multiscale modeling, the primary challenge faced by the simulation-based multiscale design community is to effectively and efficiently *utilize information* generated by wide range of models that predict system behavior at different scales. For example, as shown in Figure 1-1, the product is designed at the overall system level and individual components, through nanoscale interactions and atomic level chemistry. The objective is to achieve desired performance at the system level (e.g., an aircraft). Overall system performance is a function of its component behavior, which depends on the material properties. Material properties in turn depend on micro-scale interfaces between its constituents particles. Properties at micro-level particle interfaces depend on nanoscale interactions between molecules, which further depend on inter-atomic interactions. This shows a clear dependency of overall higher scale performance on the lower scale phenomena. Hence, all these dependencies have to be considered while designing products at multiple scales. This results in a *greater coupling* in the design, thereby increasing problem complexity. Although complexity of design is a challenge in multiscale design, the advantage of designing products at multiple scales is *increased design freedom* (i.e., a greater flexibility in configuring the system to achieve desired behavior), which enables designers to achieve performance that was not possible before. In order to illustrate this, we present a classic example of multiscale systems – *the environment* where we live.

Advantage of designing products at multiple scales: Increased design freedom (i.e., a greater flexibility in configuring the system to achieve desired behavior), which enables designers to achieve performance that was not possible before

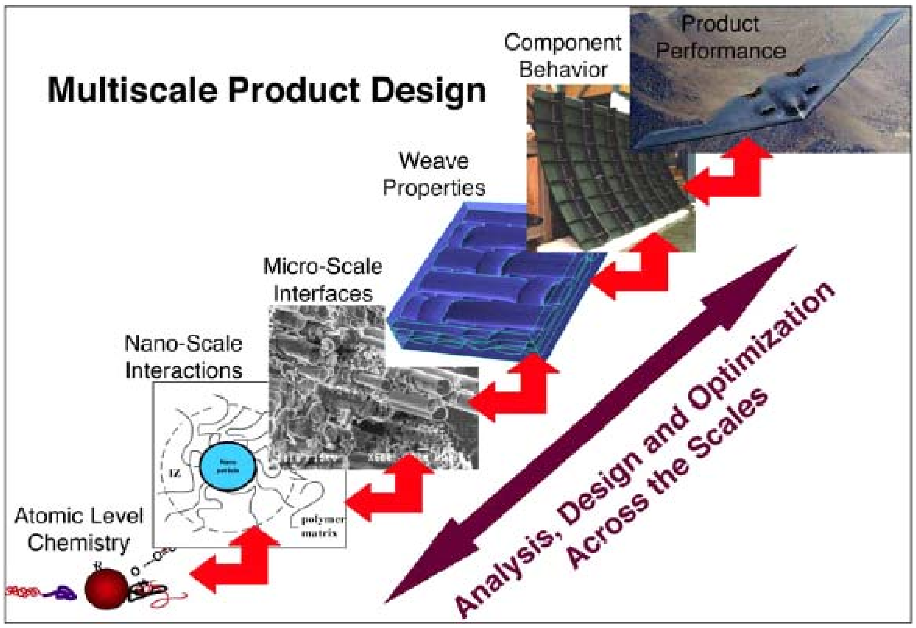


Figure 1-1 – Multiscale product design (SBES Workshop Report 2004)

1.1.1 Example of Multiscale System – The Environment

One of the highly complex multiscale problems is modeling the environment for weather prediction and to predict the human impact on climatic changes. The need for modeling climate arose from the realization that human activities can fundamentally impact earth's climate, which ultimately affects economies and natural ecosystems. Hence, it has been realized that major policy decisions should be made by considering the impact on environment. The complexity of modeling the environment stems from the presence of phenomena at length scales ranging from the order of distance between molecules (angstroms) to the diameter of the earth (several thousand kilometers). In addition to the range of spatial scales, the phenomena span a wide range of temporal scales – from fast chemical reactions to the life of earth. The processes in the earth system and their interactions are shown in Figure 1-2, which is commonly called the Bretherton diagram. In this figure, the manner in which human actions affect the climate changes is depicted.

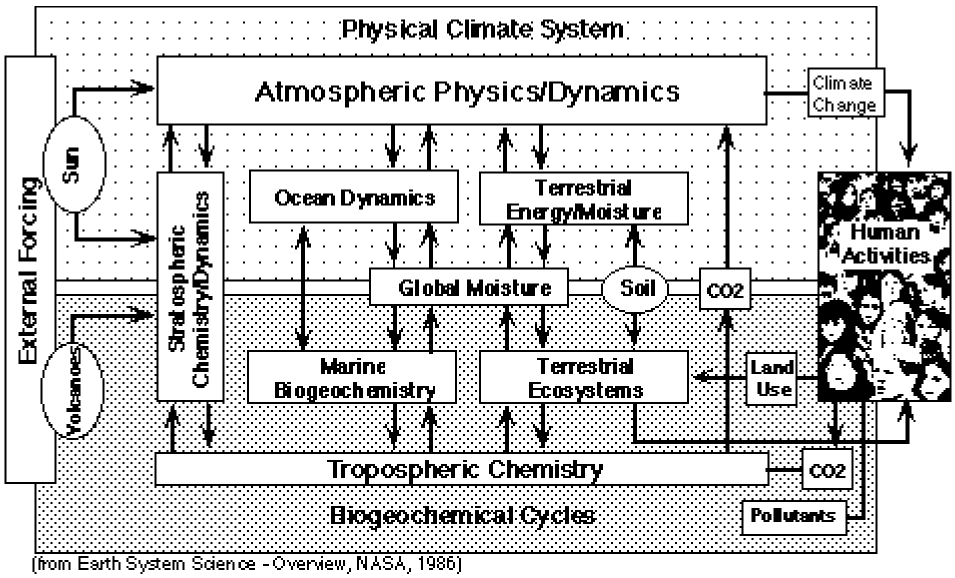


Figure 1-2 – Earth system processes (Bretherton diagram)[1]

The multiscale climate model can be used to study environmental phenomena such as sea level rise, global temperature increase, local weather prediction, ozone depletion and oxidant formation in polluted urban areas, global warming prediction, global elastic response, and seismic wave propagation. Using the model, the effect of various stimuli such as CO_2 emission and utilization, particulate emissions, on the environment can be understood. Models of the environment at different scales have been developed independently and are effective at predicting the behavior at corresponding scales. For

[1] Image taken from http://www.iitap.iastate.edu/gccourse/system/images/images.html

example, as shown in Figure 1-3, there are at least three scales of models currently available: *1)* climate level simulation, *2)* mesoscale (regional models), and *3)* microscale models. These models are briefly described next.

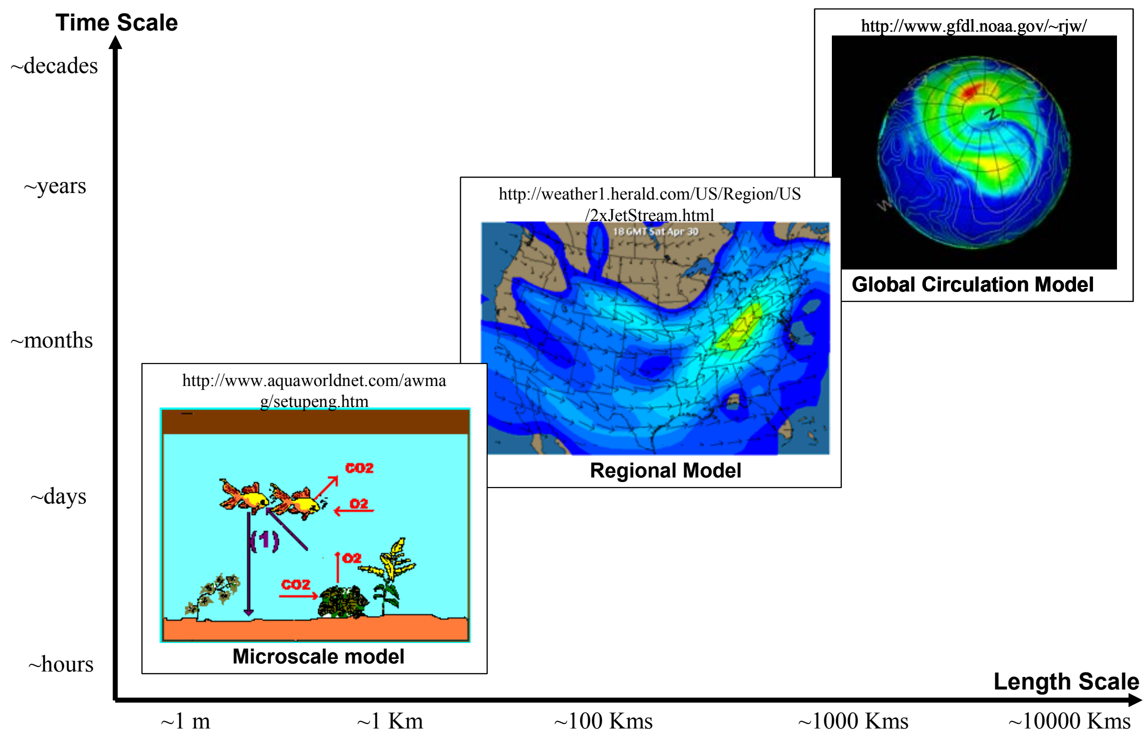


Figure 1-3 – Multiple spatial and temporal scales in the climate prediction model

1. *Climate Level Simulation (Global Circulation Model):* The components considered at the climate level simulation include *a)* the atmosphere, *b)* the oceans, *c)* the terrestrial and marine biosphere, *d)* the cryosphere (sea ice, seasonal snow cover, mountain glaciers and continental scale ice sheets) and *e)* the land surface (Houghton, Filho et al. 1997) (see Figure 1-4). These components interact through exchange of energy, water, cycling of gases (such as carbon dioxide, methane, etc), nutrients etc., and these interactions

determines the earth's climate. Different models developed for climate prediction model different number of these components and their interactions. Increasing the number of components considered increases the complexity of climate models. If a model contains enough components to effectively predict the impact on climate, it is called a *climate model*. These models are also referred to as General Circulation Models or Global Climate Models (GCMs). GCMs are based on the assumption that there is a balance between pressure gradient force and gravity. Due to this assumption, it is difficult to model phenomena with rapid changes such as concentrated downpour, downburst etc. The resolution of models at this level is of the order of 300-500 kms and the models are run for longer periods of time (greater than 30 years).

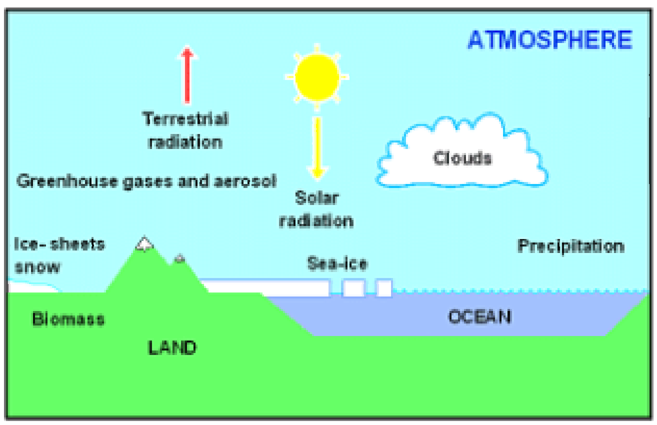


Figure 1-4 – Components of the climate system[2]

At this level, various phenomena such as cloud formation and cloud interactions with atmospheric radiation, aerosol dynamics and light scattering, ocean plumes and boundary layer, turbulent eddies, terrestrial biosphere growth, decay and species interactions, etc. cannot be modeled because these phenomena occur at scales smaller than the model resolution. Since all these

[2] Image taken from http://www.metoffice.com/research/hadleycentre/models/climate_system.html

phenomena also have a significant impact on the overall climate, they are modeled at smaller scale models such as mesoscale models.

2. *Mesoscale (Regional) Models:* Regional climate models (Meso Inc. 2005) are used to model phenomena and components at length scales (around 50kms) smaller than the Global Climate Models (GCMs) (MetOffice 2005). This is due to that fact that due to the larger resolution of GCMs, they are incapable of modeling local climate changes that are influenced by local features such as rivers, mountains, vegetation, etc. At this scale, the hydrostatic approximation is not valid. The models at this scale are based on solving three dimensional Navier Strokes equation without hydrostatic assumption.
3. *Micro Level Model:* Both the climate level and regional models do not capture the interactions between different plants, soil, chemicals in the environment, living organisms (human beings, insects, pathogens, microbes, etc.) (Takle and Kao 1998). The plants interact with soil by extracting moisture, absorbing nutrients and carbon dioxide, thereby affecting the physical phenomena at higher scale models. Hence, modeling the interactions between plants and their environment is important for understanding the behavior at higher scales. Similarly, at this level, the human influences such as cropping strategies, management practices, use of fertilizers, ultimately contribute to the overall regional and global behavior.

Although these three models at different scales are developed for simulating the behavior at corresponding scales, it is important to integrate these models in a physically meaningful manner in order to obtain a holistic understanding of the environment.

Currently, there are two commonly adopted techniques for linking these multiscale models – a) parameterization and b) concurrent coupling.

a) Parameterization is a technique through which the information from lower level models is captured into a set of parameters and their values. The parameters can be empirical or semi-empirical and can be used to approximate average behavior of physics at a lower scale. For example, at the climate level simulation, where the typical horizontal resolution is hundreds of kilometers, lower scale activities such as localized storms, clouds and land surface variations are modeled as average parameter values. All models contain parameterization at certain level. No model is capable of simulating a phenomenon *completely* using first principles. The advantage of parameterization is its resulting simplicity in accounting for the phenomena at lower scales; the disadvantage being its low accuracy. A parameter passed from one level to another can either be viewed as *a)* a property of one level to be used on another level, or *b)* a constraint arising from one level, imposed on the other level.

b) Coupling refers to the technique of using the model at one scale "on-the-fly" while performing calculations using model at another scale. Many climate modeling efforts are focused on increasing the resolution by reducing the minimum feature size modeled in the multiscale model. This requires dynamic utilization of many levels of lower scale models in the overall simulation. This requirement directly translates to the need for high performance computing tools. Hence, the ability to couple models at multiple scales is mainly dependent on the power of computational tools available. The advances in climate modeling are greatly dependent on the development in computational power, which is clear from the development of the world's No. 1 super computer (ranked in 2004) – the

Earth Simulator. Using the Earth Simulator, the resolution of climate level simulations has improved from the common 500 kilometers down to 25 kilometers, thereby generating more accurate descriptions of the underlying physics. Coupled links between multiscale models render the overall model more accurate, but at the cost of increased complexity and computational cost.

It is important to realize that even with the most sophisticated super computers available today; there is a limit on the complexity of problems that can be solved. Hence, there is a need for appropriate combination of parameterization and coupling while linking multiscale models, such that there is a balance between accuracy and computational cost. The basic question that environment modelers need to answer is – "*How much detail is required in modeling the climate?*" Currently, this question is answered by modelers based on their experience and insight into the problem. Hence, climate modeling is as much an art as a science (Houghton, Filho et al. 1997). This challenge is common across all multiscale systems. Before going into the details of each of these challenges we would like to ask ourselves the following question - "**How are multiscale systems different from conventional complex systems?**" This is a valid question because any conventional complex system such as an automobile, an airplane, a satellite, etc. spans multiple length scales – complete systems are at the order to few meters and the smallest components such as electronic sensors at the order of few millimeters (or even microns). The key difference between emerging multiscale systems and the conventional systems is the focus of the following Section 1.1.1.

1.1.1 Conventional Complex Systems vs. Multiscale Systems

It is not an exaggeration to say that most of the engineering design problems are multiscale in nature (Weinan and Engquist 2003). For example, the design of a car involves design of systems such as engine, transmission, cooling, body, etc. and their integration. Each of these systems consists of sub-systems and components that interact with each other. These *complex hierarchical systems are defined by Koch through a hierarchical structure of system, subsystem, and component level information, for which compatible solutions are sought concurrently* (Koch 1997). The difference however, is that in multiscale systems, the coupling is between physical phenomena at different scales for the same component, whereas, in hierarchical systems considered so far (such as the one considered by Koch in (Koch 1997)), the coupling is primarily between subsystems (physical components). The complexity in such conventional multiscale systems is due to coupling between components at the same level (scale). This is referred to as *horizontal coupling* (see Figure 1-5). However, the scales are not tightly coupled with each other, thereby allowing for independent design of components that can be used in the system level design. The complexity in multiscale design arises when *a)* these scales are tightly linked with each other, i.e., *vertical coupling* (see Figure 1-5); *b)* each scale is described by a different set of physical principles. Hence, in conventional hierarchical systems, the coupling is simpler to consider as compared to multiscale systems. In hierarchical systems, the challenges include integration of multiple disciplines, and require resolution of multiple conflicting objectives. The main reason for considering coupling between physics is to gain an understanding of the system efficiently and *making right decisions in an efficient manner*. The main reason for considering coupling between subsystems in a concurrent fashion is to make right decisions. The coupling is handled in hierarchical systems by introducing *intermediate responses*, *linking variables*, and *compatibility constraints*.

Complex multiscale systems of the future should be designed by considering both horizontal and vertical coupling. Although independent analyses can be carried out individually at each scale, new physical insight is developed by coupling these scales. Specific methods are developed for solving domain specific multiscale design problems but a domain independent structured methodology for multiscale systems design is not available. *Current methods in simulation-based design do not encompass the full set of performance criteria to produce better designs considering variables from all scales such as material microstructure through overall system* (SBES Workshop Report). In order to address this challenge, there is a need for domain independent methodology for designing multiscale systems.

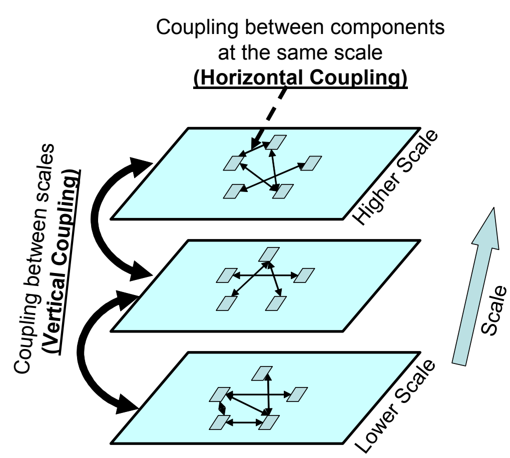


Figure 1-5 - Horizontal and vertical couplings in multiscale systems

All complex systems are characterized by three types of couplings: *a)* between components of the system, *b)* between physical phenomena, and *c)* between different scales (see Figure 1-6). The strengths of each of these couplings are different in different systems. Some of the couplings are weak and may be ignored during modeling and design, while others are strong and must be considered. *Science abounds with examples of multiscale systems in which the scales are only weakly coupled. Were this not so, we would have made little progress in the theoretical sciences* (Rudd and Broughton 2000). In the conventional systems, the strength of coupling between system components is high and the strength of couplings between physical phenomena and across different scales is weak. In the multiscale systems such as the environment, all the three types of couplings are strong and must be explicitly modeled and accounted for decision making.

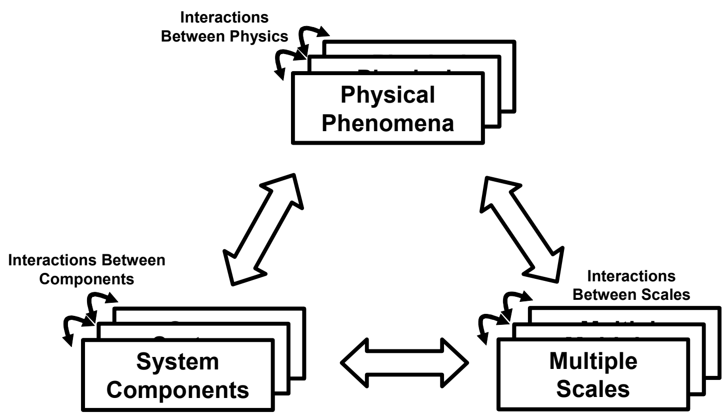


Figure 1-6 – Complex systems with multiple components, scales and physical phenomena interacting with each other

Multiscale System: A system where couplings between a) system components, b) physical phenomena, and c) scales are strong and must be explicitly modeled and accounted for decision making

Multiscale Model: A system level model that is developed by integration of information from models available at multiple scales to gain a holistic understanding of the system

Multiscale Design: Engineering field which involves accounting for all aspects of systems from lower scale materials to larger scale systems throughout the product lifecycle in order to make risk-informed design decisions at all scales

In this dissertation, a clear distinction is made between multiscale modeling and design. *Multiscale modeling* deals with efficient integration of information from models available at multiple scales to gain a holistic understanding of the system, whereas *multiscale design* deals with efficient utilization of information to satisfy design objectives. The challenges in modeling multiscale systems and existing frameworks for modeling multiscale systems are discussed in Sections 1.1.2 and 1.1.3 respectively. Section 1.1.4 is focused on the challenges associated with design of multiscale systems.

1.1.2 Challenges in Modeling Multiscale Systems

A *multiscale model* is defined as a model that takes advantage of information from various scales present in the system in order to gain a better understanding of the system while reducing the computational cost. In Section 1.1.1, we discussed an example of multiscale system - the environment is presented. Other domains that constitute multiscale systems include material science, combustion, telecommunication networks, biology, etc. Successful development in these areas must overcome the challenges described in Table 1-1 and discussed in detail in this section. Although these challenges are common to all multiscale systems, we describe these with examples environment modeling example.

1. As discussed in the previous section, one of the primary challenges in multiscale systems is to balance the need for accuracy and computational cost. Generally, the use of smaller scale model for predicting the performance of complete system provides a more accurate representation of the system. However, running these

smaller scale models at a large enough domain to capture larger effects is computational prohibitive.

Table 1-1 – Challenges associated with multiscale systems

Challenges in Multiscale Systems
1. Balancing the behavior prediction accuracy with computational cost
2. Modeling appropriate number of components in the system in order to faithfully simulate component interactions
3. Modeling relevant physical phenomena relevant to the system
4. Modeling interactions between scales and interfacing them such that they are physically compatible
5. Appropriate selection of models and model parameters at each scale (which models are appropriate at which scale)
6. Bridging the gap between different types of information (such as stochastic to deterministic, discrete to continuous, etc.)
7. Managing large quantities of information (statistical datasets) at different levels of abstraction
8. Managing complex, multidisciplinary models
9. Quantifying and handling uncertainty
10. Managing uncertainty propagation
11. Targeted refinement of models
12. Adaptive selection of details and resolution

Ideally, system modelers would like to model the behavior of a complete system using first principles. For example, in a theoretical sense, just by using the behavior at atomistic level, emergent properties of the whole system can be determined. The overall properties of a material can be predicted using the interaction of individual atoms. However, using the current computational capabilities, it is not physically possible to predict the behavior of the system using just the lowest scale models. Miller (Miller 2003) argues that the current massively parallel computers can handle only 0.01% of a typical aluminum grain. He also estimates that the total number of atoms simulated worldwide in the past 30 years is on the order of 10^{12} – which is only about 10% of the atoms in a single grain of aluminum. "*Multiscale modeling is a new paradigm, where a variety of mathematical models at different levels of details can be considered and the right combination of models is selected during the process of computation according to the accuracy needs*" (Weinan and Engquist 2003). This is the first challenge in modeling multiscale systems - *balancing the behavior prediction accuracy with computational cost* (see Table 1-1).

2. Multiscale systems often consist of multiple components. For example, in the climate level simulation, the components include oceans, atmosphere, sea ice, mountain glaciers, terrestrial and marine biosphere, etc. All these components interact with each other. Different climate models consider different number and type of components. By reducing the number of components and interactions modeled in the system, both the number of required calculations and the model fidelity reduces. Hence, modelers at each scale must *consider the appropriate components in the system in order to faithfully simulate component interactions* (see Table 1-1).

3. *Modeling required physical phenomena relevant to the system:* Most current multiscale systems are also multi-physics in nature. For example, the climate prediction model consists of the exchange of energy between various components, the cycling of gases, cloud formation, species interaction, etc. These physical phenomena are governed by different physical laws and mathematical equations. These phenomena may either be dependent on or coupled with each other. The impact of considering different phenomena is different on the accuracy of the overall system behavior prediction. Hence, in order to gain a reasonable understanding of the system, it is important to model appropriate phenomena that are related to each other.
4. *Modeling interactions between scales and interfacing them in a physically compatible manner:* In multiscale systems, phenomena at different length and time scales are generally modeled with different sets of physical laws, mathematical equations, and parameters by different domain experts. The assumptions at different levels are also different. Each of these models provides different kinds of insight into the system behavior and hence, must be integrated in a manner such that the overall model provides consistent behavior of the system. Integration of models at different scales requires consistent mathematical and physical description of coupling between scales. The challenge is that various scales depend on each other, which makes it important to determine how the different scales are linked with each other. Hence, as Rudd and Broughton (Rudd and Broughton 2000) point out, "any successful multiscale model must faithfully reproduce the intertwined nature of length scales". The ability to simulate

complete systems requires faithfully modeling how the system is connected and controlled at all the levels (Dolbow, Khaleel et al. 2004).

In summary, multiscale systems represent a special type of complex systems, characterized by multiple components, multiple physics, and multiple scales (see Figure 1-6). Appropriate modeling of all these there aspects and their interactions (between components, physics, and scales) is the key to multiscale modeling. The properties of such complex systems cannot be predicted merely through determination of individual sub-system properties. It is important to model the interactions between physical phenomena at various scales.

5. *Appropriate selection of models and model parameters at each scale (which models are appropriate at which scale):* Simulation models can be developed at different fidelity levels by changing the scope and the assumptions underlying the model. For example, a system may be modeled in one-dimension, two-dimensions, or three dimensions. Depending on the system under consideration, different models may be appropriate for predicting its behavior. It is important to select the right set of models and assumptions. It can be argued that this is a rather general requirement for any kind of simulation model development. However, the requirement is important for multiscale systems because it has to be considered at multiple scales with information from one model feeding into another. The appropriateness of models also depends on the compatibility between assumptions made in different models. Hence, a related requirement for multiscale modeling is *resolution of model mismatch to ensure compatibility between models*.

Appropriate selection of models has great impact on the accuracy and the time required for executing the models.

6. *Bridging the gap between different types of information* (such as stochastic to deterministic, discrete to continuous, etc.): Significant amount of information generated in multiscale models from different sources is generally available in different forms such as graphs, images, text, numerical and experimental data, etc. Mathematical (and software) bridges across levels of lengths and type such as stochastic to deterministic, discrete to continuous (Weinan and Engquist 2003; Dolbow, Khaleel et al. 2004) are required to integrate information from different scales.
7. *Managing large quantities of information (statistical datasets) at different levels of abstraction:* "*At each finer scale, a more detailed theory has to be used, giving rise to more detailed information about the system*" (Weinan and Engquist 2003). In addition to mathematical challenges in coupling information at different scales, the integration also needs to be carried out at a software infrastructure level. Issues such as synchronization of information generated by models at different scales, long run times, load balancing, capturing information at various levels of abstraction in a consistent database, integration of distributed computational models and hardware resources are pervasive in multiscale modeling. This has led to a new field of multiscale information science (Dolbow, Khaleel et al. 2004).
8. *Managing complex, multidisciplinary models:* In addition to managing the data, management of simulation codes is also important. A repository of simulation models from which the designers or analysts can extract the models appropriate

for their needs. The simulation model repository should be developed considering the issues such as capturing assumptions, range of validity, and model's context are important while developing simulation-based design framework for multiscale systems.

9. *Quantifying and handling uncertainty:* Any simulation model is associated with some amount of uncertainty. Uncertainty in simulation models is categorized as aleatory and epistemic uncertainty. Aleatory uncertainty refers to the uncertainty due to the inherent randomness in the physical processes, whereas epistemic uncertainty refers to the uncertainty due to lack of knowledge about the system, which can be due to lack of information about model parameters and approximations in the model. In order to make appropriate use of information generated by simulation models, uncertainty quantification plays an important role. Capturing the information about range of validity of models is also important. Uncertainty is especially important in multiscale models due to the interactions between phenomena at different scales and quantification of this uncertainty in the models is difficult. For example, according to the IPCC third assessment report (Watson, Albritton et al. 2001), the temperature projections using the SRES emissions scenarios in a range of climate models result in an increase in globally averaged surface temperature of 1.4 to 5.8°C over the period 1990 to 2100. This temperature range is large considering the impact of a few degrees increase in the global temperature.

10. *Managing uncertainty propagation:* In addition to quantifying the uncertainty in multiscale simulation models, uncertainty propagates from one model to another

along with the information flow between models. The uncertainty may either get amplified or may remain under bounds. If the uncertainty gets amplified while passing information from one model to another, the overall system level simulation model may not be acceptable although the uncertainty bounds of individual models are acceptable.

11. *Targeted refinement of models:* The accuracy of the overall multiscale simulation model is dependent on the accuracy of constituting models at individual scales and the manner in which uncertainty is amplified due to information flow from one model to another. Hence, in order to improve the accuracy of the overall model, it is important to identify the model that has the maximum impact on the overall uncertainty and then refine that model in a targeted fashion (i.e., the most critical link in the model chain and improve that link).
12. *Adaptive selection of details and resolution:* Although uncertainty is an important aspect of multiscale modeling and it should be controlled, many multiscale models can be simplified significantly reducing the model execution time without reducing the accuracy. The objective of multiscale modeling is to exploit such scenarios and to select appropriate levels of detail in the models.

As a summary, the objective of multiscale modeling is to take advantage of multiple scales in order to gain a holistic understanding of the system. The key in multiscale modeling is interactions between models at different scales. As suggested by Rudd and co-authors, "no one of those models alone would suffice to describe the entire multiscale system, but it may be possible to combine the models of different scales, effectively concentrating the computational power where it is needed the most"(Rudd and Broughton

2000). Hence, during multiscale modeling, there is a tradeoff between the computational resources and overall model fidelity due to integration of knowledge from multiple scales. As discussed in this section, some of the key challenges faced in SBES include lack of methods for bridging various time and length scales, management of models and uncertainty associated with them, management of huge amount and variety of information, and methods for efficient decision making based on the available models. Although efforts have been made to address some of these challenges for individual application domains, a domain independent framework for addressing these challenges associated with multiscale problems is not currently available in the literature. Some of the approaches currently used for multiscale modeling are discussed in the following Section 1.1.3.

1.1.3 Multiscale Modeling Approaches

It is evident from Section 1.1.2 that the main challenge in multiscale modeling is to reduce the complexity and recognize the simplicity of multiscale problems in order to generate a system description that is accurate enough for the problem at hand. In the past, there have been a number of efforts in modeling the scales individually, but multiscale modelers have shown that by appropriate combination of models at different scales, it is possible to gain a holistic understanding of the system. *Multiscale modeling employs models at different physical scales, to build a comprehensive description of systems that could not be modeled otherwise* (Rudd and Broughton 2000). Weinan and co-authors (Weinan, Engquist et al. 2003) at Princeton University categorize multiscale modeling methods into *classical* and *modern* techniques. Classical techniques refine the macroscale model using microscale models. In other words, classical techniques are essentially microscale solvers applied to the macroscale domain of interest. Examples of classical multiscale techniques include multigrid method, domain decomposition, wavelet-based methods, adaptive mesh refinement, fast multipole method, and conjugate gradient method. However, the modern multiscale methods utilize the microscale models only in the domain where it is required. For example, during the analysis of fracture, molecular level microscale model is important only at regions close to crack-tip. The regions far away from the crack tip can be modeled using macroscale (continuum) models. This results in a more efficient multiscale model, which is a combination of the microscale and macroscale model. Examples of modern multiscale models include Car-Parrinello method, Quasi-continuum method, Heterogeneous Multiscale Method (HMM), Gap-Tooth scheme, Coarse-Grained Monte Carlo Models, and Adaptive Model Refinement.

Although these techniques for multiscale modeling have proven successful in providing a greater understanding of multiscale problems by increasing their accuracy, these techniques are developed for, and employed in very specific applications. This is mainly because these multiscale models are based on specific insights into the coupling between different scales. This coupling is not only problem dependent but also dependent on the models used to describe the physics at these scales. Specific methods for coupling atomistic and continuum models are characterized by Miller (Miller). Miller argues that the main challenge in coupling the atomistic and continuum models is to model the transition region between the two domains. The multiscale methods are different depending on the way in which they model this transition region. These models include the FEAt (Finite Element and Atomistic) method, the QC (quasicontinuum) method, CLS (coupling of length scales) method, CGMD (Coarse-Grained Molecular Dynamics) method, and CADD (Coupled Atomistic and Discrete Dislocation) method. Rudd and Broughton (Rudd and Broughton 2000) provide an overview of coupled multiscale methods for generating accurate description of materials spanning electronic to macroscopic length scales. The authors review methods developed for seamless coupling between finite element models, molecular dynamics models, and semi-empirical tight binding, where models at different scales are run concurrently. The second approach – CGMD, which is a generalization of finite elements that pass smoothly from higher scales to molecular dynamics, as the mesh size is reduced to atomic spacing. All these methods developed in the multiscale materials research community are focused on using different models for different material regions and developing a hybrid coupled model. Once the issues of interfacing between the different regions are resolved, the overall hybrid multiscale model is computationally efficient because it uses the *right tool for the right part of the system* (Rudd and Broughton 2000).

Since all the methods discussed in this section are developed for specific applications, it is clear that a common (domain independent) mathematical framework for multiscale modeling is required to bridge the gap between heterogeneous models and information generated by them (Weinan, Engquist et al. 2003). Heterogeneous Multiscale Modeling (HMM) framework is an example of effort in that direction. The HMM framework consists of two main components – the first being overall macroscopic scheme and the second is to estimate the missing macroscopic data needed for the implementation of macroscopic scheme by solving the microscopic model locally.

Since multiscale modeling makes is possible to efficiently predict the behavior of systems at multiple scales; these multiscale models can be used for designing systems at multiple scales. For example, as shown in Figure 1-1, the product is designed at the overall system level and individual components, through nanoscale interactions and atomic level chemistry. The advantages of designing products at multiple scales include increased design freedom, which enables designers to achieve performance that was not possible before. In contrast to multiscale modeling, the primary challenge faced by the simulation-based design community is to effectively and efficiently *utilize information* generated by wide range of models that predict system behavior at different scales *for efficient decision making*. We address this challenge in Section 1.1.4.

1.1.4 Multiscale Design – The Need for Efficient Decision Making

Multiscale design refers to the engineering field which necessitates accounting for all aspects of systems from lower scale materials to larger scale systems throughout the product lifecycle in order to make risk informed design decisions at all scales. The primary design challenge due to multiscale nature of the problem is interactions between scales, which necessitates designers to appropriately account for coupling between scales that affect the ultimate behavior of the complete system. In this section, we discuss the challenges associated with designing multiscale systems.

The objective in designing is to utilize information generated by multiscale models in a goal-oriented manner to satisfy the requirements. *Current methods in simulation-based design do not encompass the full set of performance criteria to produce better designs considering variables from all scales – material microstructure through overall system* (SBES Workshop Report). Design of multiscale systems is characterized by the challenges of multiscale modeling and additional challenges associated with design exploration at different scales. The manner in which design activities are carried out, reflect a *problem solving* emphasis rather than *information gathering* emphasis. The overall objective in designing is to enable the design and manufacturing of increasingly complex products at lower cost and in less time. In order to perform multiscale design, the ability to develop multiscale models is a pre-requisite. Multiscale design poses additional challenges due to the need for design synthesis and associated decision-making. In contrast to multiscale modeling, where the tradeoff is between simulation time and accuracy, the tradeoff in multiscale design is between the satisfaction of design objectives and the computational time and cost for design. From the design problem solving perspective, the challenges in multiscale design are highlighted in Table 1-2 and discussed next.

Table 1-2 – Design specific challenges in multiscale systems

1. Increased number of design variables and coupling
2. Configuration of complex design processes
3. Decision-making under uncertainty
4. Evolving simulation models and requirements
5. Design exploration techniques
6. Distributed decision makers

1. *Increased number of design variables and coupling:* Couplings between multiscale models induce complexity in the associated design processes, which is further increased if the design is multi-functional. Multiscale, multi-functional design processes involve different domain experts with distributed simulation models. All these factors further complicate the design processes. In order to reduce the complexity of design processes, it is important that only the couplings that are

most important for decision making be considered in the design process. Hence, designers are faced with decisions related to the product but also the decisions related to design processes. These decisions constitute the meta-design and are called meta-level decisions[3]. The meta-level decisions are concerned with the tradeoff between the complexity of design processes, and the satisfaction of design objectives.

2. *Comfiguration of complex design processes:* It is clear that the processes used to design the multiscale systems have a major impact on the computational costs and design efficiency. Appropriately designed design processes can lead to better design solutions faster. Hence in multiscale design, there is a need for careful consideration of the design process. The design of design processes involves decisions at three levels – a) architecture level, where the precedence relationships between decisions and tasks are determined, b) individual decision level, where the simulation models used to generate information for making design decisions are selected, and c) at the individual model level, where the parameters, approximations, etc. associated with the simulation model are decided upon.

3. *Decision-making under uncertainty:* Uncertainty in design of multiscale systems arises from three sources – *a)* inherent randomness in the system, *b)* lack of knowledge about the system, and *c)* the error introduced in the models due to simplification of simulation models and design processes. Effective management of uncertainty involves making decisions robust to uncertainty in the simulation models and mitigating uncertainty through model refinement. Proper accounting of uncertainty is especially important in multiscale design because of the propagation

[3] *Designing design processes* will be used synonymously with *meta-design* throughout this dissertation

of uncertainty across different models and scales. Design methods for robust decision making under model uncertainty and propagated uncertainty are required.

4. *Evolving simulation models and requirements:* Further challenges inherent in multiscale design of systems include: *a)* improved fidelity of simulation models with time because of the evolving system knowledge, and *b)* the design requirements evolve with time. The objective, hence, is to utilize the models available at various fidelity levels and develop a preliminary design, which can be refined when additional knowledge about the models or the requirements is available. "The hierarchical nature of multiscale models offers the promise of obtaining computational improvement, especially in early stages of the optimization (design) process, by considering only as much model resolution as necessary to obtain sufficient progress at a given iteration" (SBES Workshop Report). Hence, it is important to focus the refinement effort on aspects that have the most impact on the final satisfaction of design requirements. Hence, design methods for multiscale systems should be open to refinement of simulation models.
5. *Design exploration techniques:* Multiscale problems are generally characterized by an increase in number of parameters that can be modified to achieve design goals. This increases the efforts for design exploration using conventional techniques. This calls for the development of faster and more efficient design exploration techniques. These include design of computer experiments, approximation techniques, etc. Further, multiscale design problems are characterized by long simulation runtime and large degrees of freedom. Using such models in the design

exploration loops is computationally prohibitive. Hence, efficient design of experiments and meta-modeling techniques are required to create simplified mathematical relationships between the design variables and responses that can be used for design space exploration.

6. *Distributed decision makers:* Geographical distribution of designers adds to the complexity of design processes by increasing the bandwidth of information transfer and associated design time. The distribution of design expertise and functional knowledge dictates the way in which design problem must be partitioned. In such cases, the design processes are defined not only based on the physics based coupling between parameters but also on designers' expertise and how effectively they can exchange information. This necessitates development of design methods that account for such organizational and geographical considerations.

All these design related challenges can be summed up into the following requirement – **"domain independent framework for simulation-based design of complex, multiscale, multifunctional systems"**, which is identified as the primary requirement to be addressed in this dissertation. Due to the potential breadth of multiscale modeling and design, it is important to narrow the scope of challenges to be addressed in the dissertation. We believe that one of the main aspects of such a framework is consideration of decisions related to both products and design processes. Hence, in this dissertation, we focus only on the design of design processes (meta-design) for the simulation-based design of complex multiscale, multifunctional systems. In the following

Section 1.1.5, we define our perspective and boundaries within which the requirement is addressed in this dissertation.

1.1.5 Multiscale Design Focus in this Dissertation – Designing Design Processes

The primary requirement of domain independent framework for simulation-based design of multiscale systems gives rise to the following primary research question for this dissertation:

> ***Primary Research Question*** – How should simulation-based design of complex multiscale, multifunctional systems be carried out?

This is a very broad question, and can be answered in many different ways. The question that comes up is – *What is unique about the way this question is answered in this dissertation?* In this dissertation, we adopt a unique perspective for answering this broad question. The primary hypothesis used in this dissertation to answer this question is that "***simulation-based design of multiscale, multifunctional systems can be carried out by decision-based integrated design of products along with their design processes***". Notice that the existing Integrated Product and Process Design (IPPD) efforts are also focused on the integrated design of products and processes, but the processes in the context of IPPD are *manufacturing processes* rather than design processes. The key aspects of this hypothesis include: *a)* integrated design of products and design processes, and *b)* decision-based design.

The first key aspect of the primary hypotheses is "*integrated design of products and design processes",* which represents a shift from the traditional view of design where the design processes are first formulated based on the functional decomposition and previous experience and the focus is mainly on designing the products. In contrast to that, we advocate systematic design of design processes (meta-design) in conjunction with the products because we believe that for complex design scenarios, design processes have major impact on the final outcome and the efficiency with designers arrive at the design solution. Although elements of designing design processes can be found in different efforts, a *general framework* for performing integrated design of products and design processes is lacking in the design literature. The dictionary definitions of a framework are: "a fundamental structure that supports something", "a set of assumptions, concepts, values, and practices", "a basic conceptional structure (as of ideas)". In this dissertation, our goal is to develop a fundamental conceptual structure that supports designers in performing integrated design of products and design processes. This fundamental structure consists of a set of theoretical constructs, methods, approaches, metrics, and tools. Domain and application independence is one of the key characteristics of this framework. The objective is to develop this framework based on well established design methods. The objective is not to develop a fixed set of steps that will solve the problems in a particular domain, but a general enough approach that can be extended and particularized for domain specific problems. The general approach adopted in this dissertation for integrated design of products and design processes is the assumption that both multiscale systems and design processes are special types of hierarchical systems. Hence, designers can apply concepts from product-design to the design of **both** a) multiscale systems and b) design processes. It is important to note that although these concepts are developed and applied in this dissertation for multiscale systems, they are valid for any other system.

The second key phrase in the primary hypothesis is "*decision-based design",* which is used as a basis for the development of the design method, and the metrics for making meta-level decisions, and modeling design processes. The main aspect of decision-based design is that the key role of a designer in a design process is to make decisions. Hence, decisions represent the most important activity in a design process. All other activities are only for generating or transforming information to support designers' decision making. Developing and executing simulation models represent two such information generating activities. Decision-based design is chosen as the underlying construct because of its domain independence and the ability to integrate different perspectives into a common thread. From this perspective, the design of design processes can be viewed as a network of decisions about the design processes. Since these decisions are not about the product itself, but they have an impact on the final product, they are referred to as *meta-level decisions* in this dissertation. The meta-level decisions involve tradeoffs between the cost of making decisions, satisfaction of design requirements, and the quality of design decisions. This is analogous to the tradeoff between accuracy and computation cost in multiscale modeling as discussed in Section 1.1.2.

Further, the framework developed in this dissertation is also geared towards efficient and effective utilization of information in the design process. The word 'efficiency' in the context of decision-based design refers to the speed with which information for decision making is generated and provided to decision makers and the word 'effectiveness' refers to the quality and relevance of this information for supporting decisions about the product. Having discussed the primary research question and the primary hypothesis, we now discuss the requirements for embodying the primary hypothesis for design of multiscale systems in Section 1.1.6.

1.1.6 Requirements List for a Framework for Integrated Design of Products and Design Processes

The requirements for a framework for integrated design of products and design processes are listed in Table 1-3 and discussed next.

Table 1-3 – Requirements list for the framework for integrated design of multiscale products and design processes

Requirements list *for integrated design of product and design processes*

1. A method for integrated design of products and design processes
2. Support for decentralized, multifunctional design
3. Metrics to quantify the performance of different design process alternatives
4. Support simplification of complex design processes without affecting the performance of the product
5. Support evolving simulation models
6. Support design process exploration, and reusability of existing design process, product and decision related information and knowledge

Meta-design involves making various decisions about the design processes such as configuration of tasks for enhancing concurrency, selecting the manner in which individual design tasks should be carried out, and process parameters associated with each design task. Current popular design methods such as Pahl and Beitz (Pahl and Beitz 1996) or the Systems Engineering Vee (Buede 2000) do not provide systematic means for making these meta-level decisions. Hence, the first requirement for a multiscale design framework is to develop *a method for integrated design of products and design processes*. Decentralization of domain experts and decision makers is an important charateristic of multiscale, multifunctional design problems. Such decentralization imposes additional requirements for collaborative decision-making with limited bandwidths for information exchange. Hence, the second requirement for the framework is to *support decentralized, multifunctional design*.

As discussed in the Section 1.1.4, multiscale design scenarios are characterized with important meta-level decisions that can possibly simplify the complex design processes and associated computational costs. Hence, from a simulation-based multiscale design standpoint, there is a need for adaptive decision making about the design process that includes considerations such as:

a) the appropriate scales that should be considered,

b) the level of detail in models considered at each scale,

c) interactions that need to be modeled between models,

d) degree of parallelization,

e) interactions between decisions that should be modeled, and a finer level,

f) what level of idealizations (such as numerical discretizations) is appropriate, etc. Different decisions result in the selection of different design process alternatives. In order to make design process related decisions (or to select appropriate design process alternative), there is a need for *metrics to quantify the performance of different design process alternatives*. This is the third requirement for the framework for integrated design of products and design processes.

One of the important types of decisions about design processes is related to determining the *appropriate* level of detail to be considered in the design process such that the design objectives are satisfied with as little simulation effort as possible. In other words, the design framework should *support simplification of complex design processes while satisfying the desired performance requirements*. This is the forth requirement for the design framework. The fifth requirement for the design framework is related to the fact that simulation models are not static – they evolve with time because of the increase in knowledge about the system behavior along the design process, and rapidly increasing ability to model complex phenomena. The design framework should be capable of *supporting evolving simulation models*. In design, it is also important to determine the right level of refinement of simulation models, beyond which, gathering additional information is not advantageous from the decision making perspective.

In addition to the development of design methods incorporating meta-design and design, there is a need to implement these methods in a computational framework to support meta-design. All the current design frameworks are focused on the design phase only. Currently available design frameworks do not address the configuration of design processes. Hence, the design framework should also include computational tools to *support design process exploration*, which is the sixth requirement for the design framework. The computational tools should also support reusability of information and knowledge gained in previous design scenarios to new design problems. The reusability should not be limited only to product related information. It should extend to the information about products and design decisions. This is the sixth requirement for design framework for a framework supporting integrated design of products and design processes. The requirements for the framework for integrated design of products and design processes are summarized in Table 1-3.

These six requirements are categorized into three general research areas: *a)* methods for integrated design of products and design processes, *b)* metrics for analyzing design processes, and *c)* modeling design processes to support meta-design. In order to address the requirements discussed in this section, a design framework is proposed in this dissertation. This design framework is composed of three building blocks as shown in Figure 1-7 including *a)* a robust multiscale design exploration method, *b)* value of information based metrics for meta-level decisions, and *c)* an information modeling strategy.

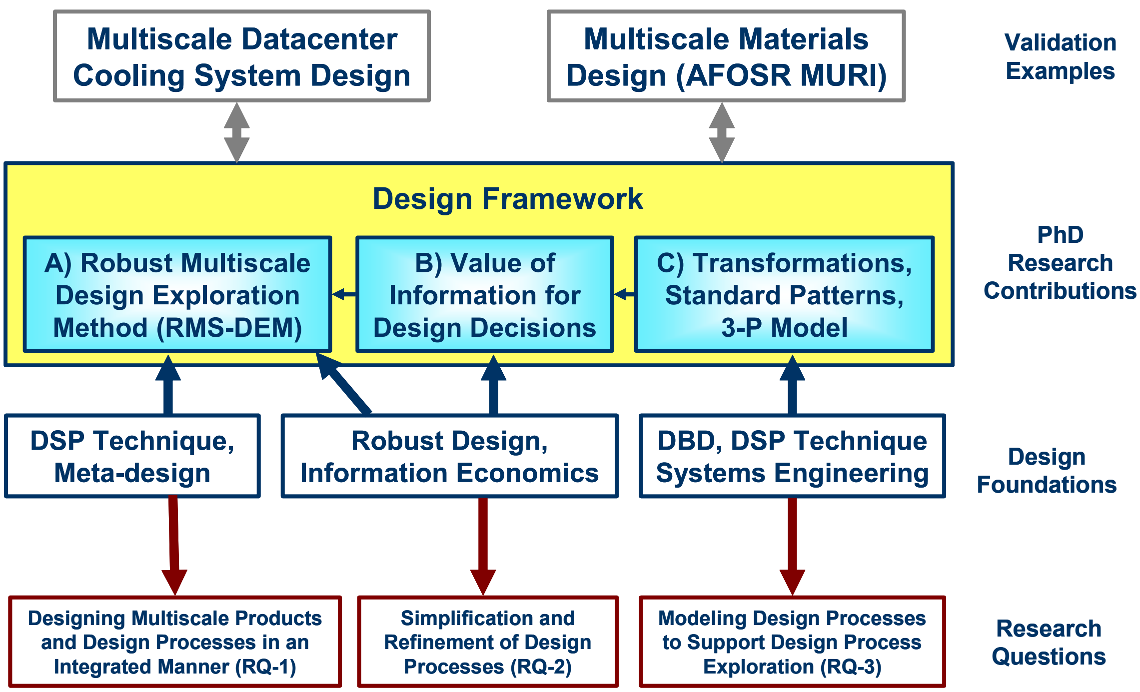


Figure 1-7 – A summary of research questions, design foundations, research contributions, and validation examples used to validate the contributions in this dissertation

The design framework is based on existing design foundations such as decision-based design, Decision Support Problem (DSP) Technique, systems engineering, robust design, and information economics. The details of these foundations are discussed in Chapter 2. The three research areas are associated with three research questions that are discussed in detail in the following Section 1.2. The hypotheses for answering these research questions are introduced and an overview of the research contributions is also provided in the following section.

1.2 Answering the Primary Research Question – Identifying Research Questions and Corresponding Hypotheses

Each of the three research areas identified in the previous section is discussed in detail in Sections 1.2.1, 1.2.2, and 1.2.3. From these research areas, three research questions and associated hypotheses are discussed. A mapping of the five requirements for the framework to the three research questions is provided in Table 1-4. The hypotheses used to answer the three research questions and associated contributions from the dissertation are highlighted in Table 1-5 and discussed in the following.

Table 1-4 - Mapping the framework requirements with research questions

Requirements for a framework for integrated design of products and design processes	Research Questions for this dissertation
1. A method for integrated design of products and design processes	**RQ 1.** How can simulation-based multiscale design processes be designed in association with products?
2. Support for decentralized, multifunctional design	
3. Metrics to quantify the performance of different design process alternatives	**RQ 2.** How should multiscale design processes be systematically simplified and models refined in a targeted manner to support quick design decision making without compromising their quality?
4. Support simplification of complex design processes without affecting the performance of the product	
5. Support evolving simulation models	
6. Support design process exploration, and reusability of existing design process, product and decision related information and knowledge	**RQ 3.** How should simulation-based design processes be modeled in a systematic manner and represented in a computer interpretable format to support design process exploration?

Table 1-5 – Mapping the requirements, research questions, and hypotheses

<table>
<tr><td rowspan="3">Primary</td><td>Requirement</td><td colspan="3">Domain independent framework for simulation-based design of complex, multiscale, multifunctional systems</td></tr>
<tr><td>Research Question</td><td colspan="3">How should simulation-based design of complex multiscale, multifunctional systems be carried out?</td></tr>
<tr><td>Research Hypothesis</td><td colspan="3">Simulation-based design of multiscale, multifunctional systems can be carried out by decision-based integrated design of products and design processes.</td></tr>
<tr><td></td><td>Requirements</td><td>Research Questions</td><td>Hypotheses</td><td>Contributions</td></tr>
<tr><td>I</td><td>1. A method for integrated design of products and design processes

2. Support for decentralized, multifunctional design</td><td>Q1. How can simulation-based multiscale design processes be designed in association with products?</td><td>H1.1. Systematic, step-wise refinement of design processes and the associated products increases the efficiency and effectiveness of design decision-making

H1.2. Design processes can be designed as hierarchical systems composed of repeating building blocks defined in terms of interaction patterns</td><td>Robust Multiscale Design Exploration Method (RMS-DEM)

1. A method for multiscale design of products and associated design processes

2. Application of the design method for design of materials

3. Explicit accounting of metadesign decisions in the context of designers' preferences</td></tr>
<tr><td>II</td><td>3. Metrics to quantify the performance of different design process alternatives

4. Support simplification of complex design processes without affecting the performance of the product

5. Support evolving simulation models</td><td>Q2. How should multiscale design processes be systematically simplified and models refined in a targeted manner to support faster design decision making without compromising their quality?</td><td>H2.1. Design processes can be simplified and models refined by making tradeoffs between value of information obtained via simulations and need to achieve robust, satisficing solutions

H2.2. Design processes can be simplified using decoupling of scales, decisions and functionalities</td><td>Methods and metrics for design process simplification and model refinement

1. Scale, Decision, and Functional decoupling

2. Methods for decomposing weak (robustness based) and strong (interval based) coupling

3. Value of information metric rooted in utility theory (value theory)</td></tr>
<tr><td>III</td><td>6. Support design process exploration, and reusability of existing design process, product and decision related information and knowledge</td><td>Q3. How should simulation-based design processes be modeled in a systematic manner and represented in a computer interpretable format to support design process exploration?</td><td>H3.1. Separation of product, process, and problem information enhances reusability of design process information across different products, thereby supporting meta-design</td><td>3-P information modeling approach

1. An approach for modeling design information to support meta-design

2. Reusable process patterns useful for hierarchical modeling of processes

3. Preliminary information models for Problem, Process, and Product</td></tr>
</table>

1.2.1 Research Area 1: Method for Integrated Design of Products and Design Processes

As discussed in Section 1.1.6, the first requirement is a method for the integrated design of products and design processes. Design of products has been the primary focus in engineering design research. Design methods such as Pahl and Beitz (Pahl and Beitz 1996) are based on functional decomposition and hence, the associated design processes are primarily dictated by product's functions. Design of design processes is not explicitly addressed in these traditional design methods. Some aspects of designing design processes are addressed in two research efforts – *a)* Decision Support Problem (DSP) Technique (Muster and Mistree 1988), and *b)* Design Structure Matrix (DSM) (Steward 1981; Eppinger 1991; Eppinger, Whitney et al. 1994; Eppinger and Salminen 2001; Browning and Eppinger 2002). Bras, in his dissertation developed a mechanism for designing design processes in the form of design of support problems (Bras 1992). The research is carried out as an extension of Decision Support Problem (DSP) Technique, according to which, design processes can be represented as a network of decisions that are associated with support problems. DSP Technique is an embodiment of decision-based design. Designing design processes in the context of DSP Technique refers to the design of support problems. The details of DSP Technique are discussed in Section 2.2. Bras developed mathematical models for variant, adaptive, and original design of support problems based on the compromise Decision Support Problem (cDSP) (Bras 1992). These mathematical constructs formalize the decisions made by designers during the meta-design phase. Although DSP Technique is one of the first efforts towards formalizing individual design process related decisions, the literature on DSP Technique lacks a method for systematically designing both products and design processes. Design

Structure Matrix is a construct developed (Warfield 1973; Steward 1981; Eppinger, Whitney et al. 1990) to make decisions about general processes including design processes. The focus of research in DSM is on modeling processes as networks of tasks and representing them in a matrix form. Using the matrix manipulations, the processes can be analyzed and designed to maximize concurrency between tasks, minimize coupling, reduce the overall time required for execution of the process, etc. The primary limitations of DSM for integrated design of products and design processes include *a)* inability to capture complex non-linear relationships between tasks and parameters, *b)* inability to model and analyze the impact of coupling strengths, *c)* inability to capture the impact of uncertainty in design processes, and *d)* inability to capture designers' preferences. The details of these limitations and the literature on DSM for designing design processes are discussed in Section 3.3.2. Further, similar to the limitation of DSP Technique, literature in DSM does not provide a method for integrated design of products and design processes. The DSM construct supports designers in making individual meta-design decisions only. Considering the limitations of existing literature related to designing design processes, we formulate the first research question as follows:

Research Question 1: How can simulation-based multiscale design processes be designed in association with products?

The obvious strategy for answering this research question is to identify the similarity between design processes and products, and to apply the product design methods for designing design processes. Consider an extremely simplified design method shown in Figure 1-8 that starts with identifying the product requirements and generating the design in three steps – a) generating alternatives that can satisfy the product requirements, b) evaluating the alternatives, and c) selecting the best alternative. If this simple method is applied to designing design processes, then the designers start with a set of process requirements, generate design process alternatives, and evaluate the process alternatives. The evaluation of design process alternatives is carried out based on metrics such as cost of executing the process, time for execution, and the final design outcome measured in terms of product performance. Based on the comparison of design process alternatives in terms of these metrics, the best design process alternative is selected. The selected design process is then executed to design the product.

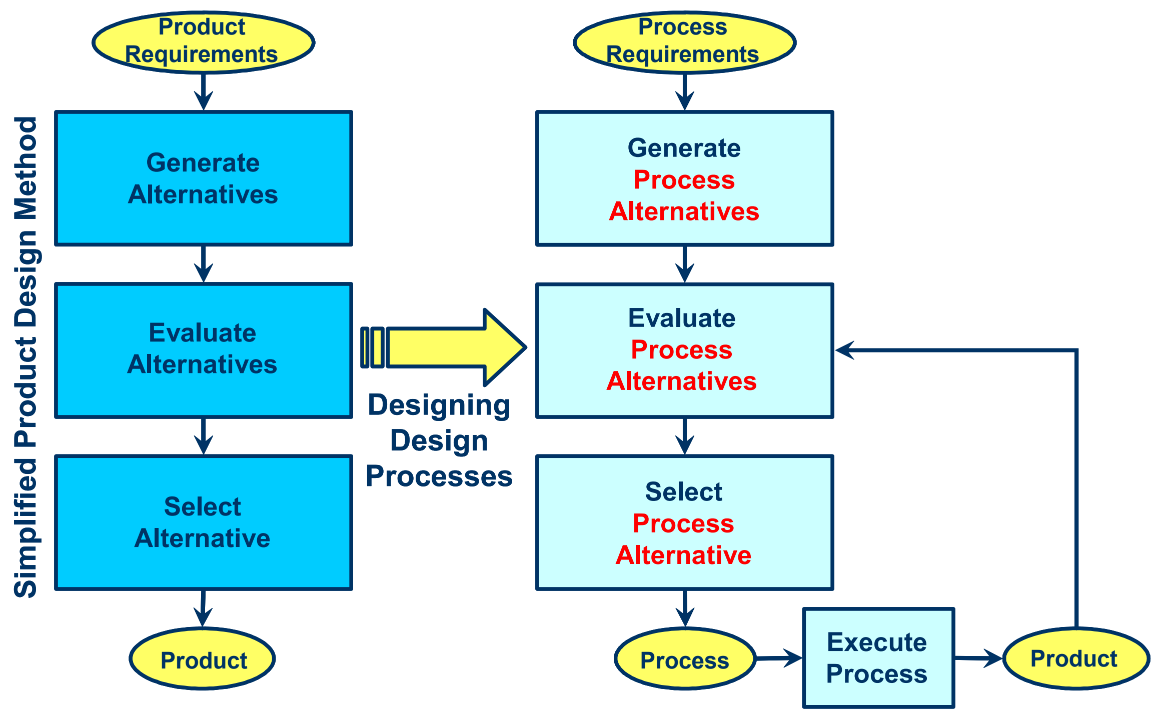


Figure 1-8 – Designing design processes in a manner similar to designing products

Although this method looks simple for designing design processes, there is one major barrier that prevents the usability of this method for designing design processes. The barrier is that the design process alternatives cannot be evaluated until they are executed.

Hence, in order to evaluate the design process alternatives, all design process options must be executed and the outcome of each process compared. If all the design process options need to be executed before one can be selected, the purpose of designing design processes itself defeated. The method for designing design processes should be such that it does not require execution of all possible options. Another barrier for designing design processes is that the design processes generally cannot be designed entirely before they are executed in the design phase. This is because the design of design processes generates information about the layout of tasks and decisions that is used in the design phase for making decisions about the products. These product-related decisions then generate information about the design processes, which can then be used for determining further details of design processes. The process of designing design processes and products must be carried out in a cyclic fashion until all the details of both are finalized.

In order to address these challenges, the proposed strategy in this dissertation is to start with preliminary design processes and then refine them systematically in association with the refinement of products. The strategy is shown in Figure 1-9, where the designers start with requirements for design processes and generate process alternatives. These design process alternatives are arranged in the increasing order of fidelity. It is assumed that the design processes with higher fidelity are more complex and take more cost and time for execution. Instead of executing all the design process alternatives and then comparing them, the designers start with a simple process alternative and execute it. The outcome of that process alternative is used to determine how much benefit can possibly be achieved using a more complex design process. If the potential benefit is significant as compared to the additional cost, the designers use a design process of higher fidelity and execute it. If the potential benefit is insignificant, then the design solution is accepted.

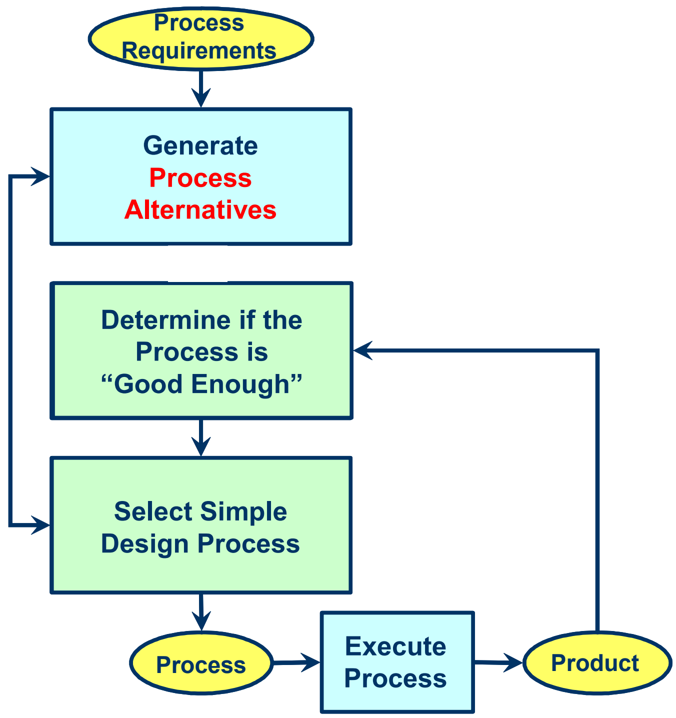


Figure 1-9 – Systematic refinement of design processes

This strategy is embodied in the first hypothesis (H1.1) for answering Research Question 1 - *preliminary design processes should be developed first, that then be systematically refined in a stepwise fashion*. In association with the design processes, the products are also refined from preliminary design to the detailed design. The word ‘systematic’ in the hypothesis refers to the fact that design processes are designed based on rigorous consideration of impact of design processes on the product related decisions. The word ‘stepwise’ refers to the fact that the process is carried out in a cyclic fashion with alternating decisions about products and processes. During the process of refinement of products and design processes, the design space is systematically reduced and the knowledge about the system progressively increases, which in turn increases the *effectiveness* of decision making. Since we are using simpler design processes in the earlier stages when the design space is large, (instead of using the most complicated design process for making decisions) the *efficiency* of decision making is also better.

The second hypothesis (H1.2) is based on exploiting the similarity between products and design processes for performing design of design processes. The starting premise is that if processes are considered as systems, their design should be similar to design of products. The design of design processes is similar to design of products if the processes are also viewed as systems. This notion of processes as systems is emphasized by Bras in his dissertation (Bras 1992) as follows "*with respect to designing a system, the definitions of system are applicable to both the object being designed and the design process itself. While designing a particular design process, this design process becomes the product of the high level design process*". Hence, the basic activities in design of products – analysis, synthesis, and evaluation should also be applicable to design of design processes. Analysis, in the context of product design refers to coming up with the behavior of the product, given the product's structure. Analogously, in the context of process design, analysis refers to determining how a given process architecture performs. The performance of processes can be measured in terms of simple metrics such as (computational) cost of execution, time required, its effectiveness, etc. Synthesis, in the context of product design, refers to generating the structure of design, given the behavior specifications. As relevant to process design, synthesis refers to generating process alternatives for performing product design. Similarly, evaluation refers to comparing the achieved behavior (of products and processes) against the desired behavior. Analysis of design processes is the focus of Research Area 2, and is discussed in Section 1.2.2. In this section, the focus is on synthesis.

Taking the analogy of products and design processes a step further, the concepts from modular product design are applied to design processes. The second hypothesis in support of answering the first question is that design processes can be designed as hierarchical systems composed of repeating (reusable) building blocks. The 'repeatability' refers to the assumption that any design process (any domain, and at any level of abstraction) can be defined in terms of a few building blocks that repeat themselves in any design process. Although these building blocks can be defined in many different ways, the focus in this dissertation is on building blocks defined in terms of different kinds of interactions between process elements. Interactions are used as the basis for defining process building blocks because we believe that by systematically considering the interactions, complex design processes can be simplified and design efficiency can be increased substantially. In other words, the repeatble interaction patterns are used to generate different design process alternatives.

The second hypothesis (H1.2) is inspired from the following question – ***"If products can be designed as modular systems, why can't we design the design processes in a similar way?"*** From a modular systems perspective, synthesis refers to the putting together components of a subsystem in order to fulfill the desired function. From meta-design standpoint, synthesis refers to putting together sub-processes together to develop a complete design process that satisfies the desired purpose (i.e., the design of artifact). This hypothesis is also related to the efficient decision making because the design effort can be reduced reduced if the process is synthesized by reusing process elements that are captured in other design scenarios.

The two hypotheses for answering the first research questions are summarized as follows:

H1.1. Systematic, stepwise refinement of design processes and the associated products increases the efficiency and effectiveness of design decision making

H1.2. Design processes can be designed as hierarchical systems composed of repeating building blocks defined in terms of interaction patterns

1.2.2 Research Area 2: Metrics for Analyzing Design Processes

From the perspective of the analysis-synthesis-evaluation paradigm applied to the design of design processes, one of the key requirements is to analyze the performance of different design processes such that better design processes can be differentiated from poor design processes. Note that the betterness of a design process refers to the effectiveness and efficiency of design processes. Hence, there is a need to develop quantitative metrics for analyzing design processes. These metrics help in evaluating different design process alternatives, thereby enabling design processes related decision-making. These metrics also help in assessing the performance of design processes with regard to design process related objectives.

Design processes can be analyzed and designed using various considerations depending on the designers' needs such as cost, and execution time, performance, etc. Although these are the preferable metrics for analyzing design processes because they directly relate to designers' preferences, the problem with these direct metrics is that the design process needs to be executed to evaluate them. Other *indirect* metrics for analyzing the performance of deign processes include concurrency, complexity, uncertainty, stability, convergence, robustness, modularity, and reconfigurability. These metrics are related to the *direct* metrics such as cost and time. For example, concurrency in design processes is directly proportional to the number of tasks that can be performed in parallel, and inversely proportional to the tasks that can be carried out in series. If the concurrency metric for a design process is increased, the time required for execution of the overall process is reduced. Hence, the objective in a design process can be to maximize the number of tasks that can be performed in parallel. Various (manufacturing) process design efforts (Kusiak, Larson et al. 1994) are based on concurrency as the metric. Complexity of design processes is proportional to the number and strength of interactions and dependencies between tasks. Greater complexity of design processes results in greater execution time due to greater coupling between tasks. Robustness of design processes refers to the ability to use the same design processes even when the design requirements change. If the design processes need to be reconfigured entirely differently on a small change in requirements, the process is not robust. Metrics for robustness of products exist (Taguchi 1986) but the metrics for robustness of design processes are not available in the literature. A modular process consists of subprocesses that can be reused over and over again. Modular design processes are preferred over non-modular processes because modularity reduces the effort involved in design them. Performance of design processes can also be measured in terms of the manner in which processes reduce the design freedom with design decisions. Design processes that help designers in keeping available design options open as long as possible are considered better than the design processes that result in faster reduction in design options. This is because if the options are open for a longer time, the adaptability to changes in environment is high.

Although these are different metrics for measuring the performance (behavior) of design processes, in this dissertation, the focus is only on metrics based on ***information economics*** for analyzing design processes. This is because our focus is on the analysis of design processes from the perspective of value of information generated by simulation models for decision-making in the design of multiscale systems. Due to the potentially broad scope of the research area of analyzing design processes, the scope of the second research question is limited to analyzing different process options that are simplifications or refinements of each other. Specifically, we compare the performance of different design processes where the main difference between the processes is: *a)* consideration of information flows between tasks, and *b)* different fidelities of simulation models used for decision making.

Ability to compare different process options based on these two aspects, using metrics, allows designers to determine the appropriate level of simplification of design processes (in terms of couplings to be considered) and the appropriate level of simulation models for decision making. Note that by considering all the couplings between design activities, and by choosing the most accurate (which is generally the most complicated) simulation model, the effectiveness of decision making is the maximum but the efficiency is low. By selecting simpler design process options and simulation models, the effectiveness of decision-making reduces. Hence there is a design process (meta-design) level tradeoff between the complexity of design processes and the quality of decisions made. In order to support designers to make design process decisions in a systematic manner under this tradeoff, the second research question is framed as follows:

Research Question 2: How should multiscale design processes be systematically simplified and models refined in a targeted manner to support faster design decision making without compromising the decision quality?

The answer to this research question consists of two parts – development of metrics for value of information and using the metric for simplification of design processes and refinement of simulation models. The premise behind using this metric is that simplification of design processes 'hides' information from the designer that may be useful for increasing the quality of designer's decision making capability. If the impact of this 'hidden' information on designer's decision is not significant, then the simplification of design processes is appropriate, otherwise not. Similarly, the refinement of simulation model adds information about the system that may be useful for designer's decision making. If the impact of this additional information has a significant impact on designer's decision making capability, then the model should be refined, otherwise not. This notion of addition/hiding of information and the quantification of 'significant impact' is embodied in the metric for value of information.

The simplification of design processes as addressed in this dissertation builds on the first research question (RQ1) and the hypothesis H1.2, according to which, design processes can be modeled using reusable building blocks defined in terms of interaction patterns. Hence, the use of metric in this dissertation is limited to the simplification of interaction patterns. The refinement of simulation models is also limited to parametric refinement. Hence, the hypotheses used to support the answer to this research question are formulated as follows:

H2.1. Design processes can be simplified and models refined by making tradeoffs between value of information obtained via simulations and need to achieve robust, satisficing solutions

H2.2. Design processes can be simplified using decoupling of scales, decisions and functionalities

1.2.3 Research Area 3: Modeling Design Processes to Support Meta-Design

The third research question is related to providing computer support for performing integrated design of products and design processes. In spite of the fundamental importance of meta-design in expending resources, it is not effectively supported by current Computer Aided Engineering (CAE) and Product Lifecycle Management (PLM) frameworks. The question, naturally arising from this observation is: "***How should CAE and PLM frameworks be developed/modified to support meta-design?***" Although this query can be posed for most design frameworks, we primarily focus on simulation-based design frameworks such as FIPER (Engenious Inc. 2004), ModelCenter (Phoenix Integration Inc. 2004), iSIGHT (Engineous Inc. 2004), etc. Such CAE and PLM frameworks adopt a tool-centric view of design processes, according to which, a design process is a network comprised of software tools, employed for processing information. The adoption of a tool centric perspective in developing design frameworks, thus invariably focuses the underlying effort on achieving interoperability between 1) different tools that perform similar function (such as different CAD applications), 2) tools providing different functionality (structural analysis, crash, vibration, etc.), and 3) applications pertaining to different domains. Various standards such as STEP, XML, and

UML are being developed to achieve interoperability between such tools. Recently, Peak and co-authors (Peak, Lubell et al. 2004) proposed a model-centric perspective to support the further development of these frameworks. Specifically, a product information model comprises a central core, modified and populated using all relevant tools. Such a model-centric view constitutes a significant improvement over the tool-centric view, commonly espoused, because information is no longer tied solely to the particular tools used for its creation or modification. A model-centric perspective is important for realizing the seamless integration of information models associated with different aspects of product design, and useful for guiding the development of CAE and PLM frameworks to support fine grained interoperability, as well as, the development of a collective product model. However, the assertion in this dissertation is that neither the tool-centric nor model-centric perspectives (alone or in concert) are adequate for effectively supporting meta-design. These two perspectives to modeling design information do not capture the fact that both design processes and products are designed in an integrated fashion. Hence, the research question for this dissertation is:

Research Question 3: How should simulation-based design processes be modeled in a systematic manner and represented in a computer interpretable format to support meta-design?

The first fundamental obstacle that prevents the design of design processes in current frameworks is the integrated manner in which information about the products and design processes is captured. The processes are actually defined 'in terms of the product information'. Hence, for a given design scenario, it is difficult to explore different design processes. The hypothesis in this research is that designers can overcome this obstacle by separating the process specific information from product specific information in a modular fashion that allows rapid utilization of different design processes for a product design scenario.

Another fundamental obstacle in furnishing the capability for meta-design is the inability of current tools to capture the problem solving aspect of design. In fact, such tools are primarily used to capture procedural aspects. Put another way, current tools do not capture *a)* what the design problem is, *b)* how the designer partitions the problem, and *c)* how different problems are related. Instead, current tools only capture the specific series of steps a designer adopts when solving the problem at hand in a quasi documentary fashion. Design problem changes thus cannot be translated to the procedural information captured within the individual tools. The word "problem" has been used in many different ways in the engineering design community. ***In this dissertation, we define a problem as "either an obstacle to be overcome or a question to be answered"***. This definition is taken from Ref. (Muster and Mistree 1988). This definition is different from the text book type problem solving, where the problem is completely defined and can be solved using a predefined set of steps resulting in a unique solution (see ref. (Hazelrigg 1998)). In real design scenarios, designers are faced with problems where complete information for solving the problem is not available and the closed form solution is not available. Without capturing the problem solving aspect of design in the CAE and PLM frameworks, it is difficult to support meta-design.

Combining the two aspects of simulation-based design frameworks discussed in this section – separation of product and process specific information, and explicit capturing of problem related information, the hypothesis for answering the third research question is summarized as follows:

H3.1. Separation of product, process, and problem related information enhances reusability of design process information across different products, thereby supporting meta-design

A summary of the primary requirement, primary research question the three research questions and associated hypotheses for this dissertation is provided in Table 1-5. The relationship between the motivation, research questions, and hypotheses is shown in Figure 1-10.

1.2.4 Research Contributions

The requirements for a framework for integrated design of products and design processes are presented in Table 1-3. In order to address those six requirements, six components of the framework are developed in this dissertation. An overview of these six components of the framework and the associated research contributions are illustrated in Figure 1-11. In this figure, the motivation of this dissertation – design of multiscale systems is shown along with two design examples: energetic-structural materials, and datacenter cooling system. The method for integrated design of products and design processes is presented in Chapter 3. The value of information metric is discussed in Chapter 4. The use of value of information metrics for refinement of simulation models is validated in Chapter 4 and Chapter 9. The use of value of information for simplification of design processes is validated in Chapter 5 and Chapter 9. The information modeling strategy is discussed and demonstrated in Chapter 7 and Chapter 8. This figure is used as a running icon throughout the dissertation to highlight the hypotheses and validation addressed in the corresponding chapters.

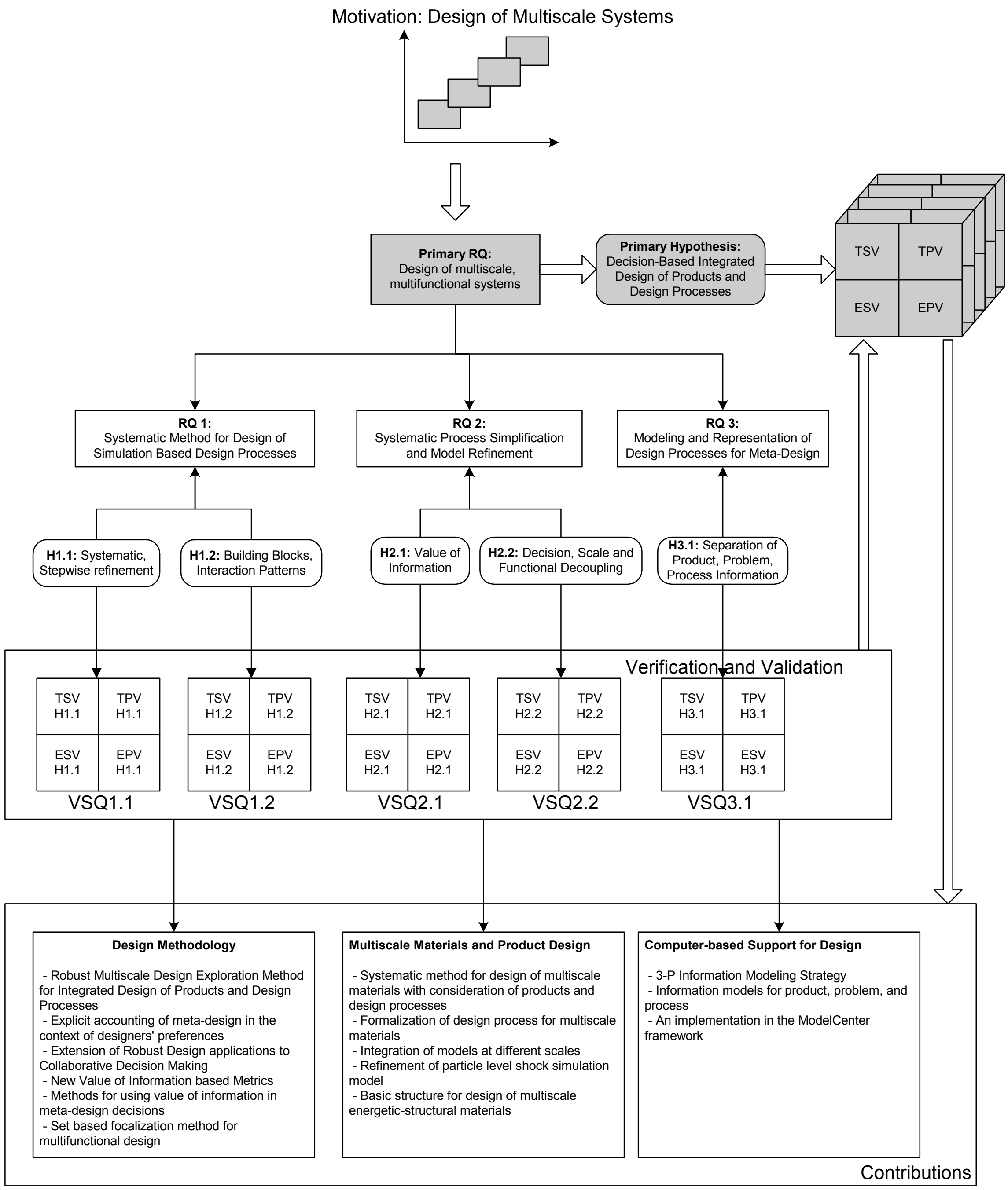


Figure 1-10 – An overview of the research questions and hypothesis

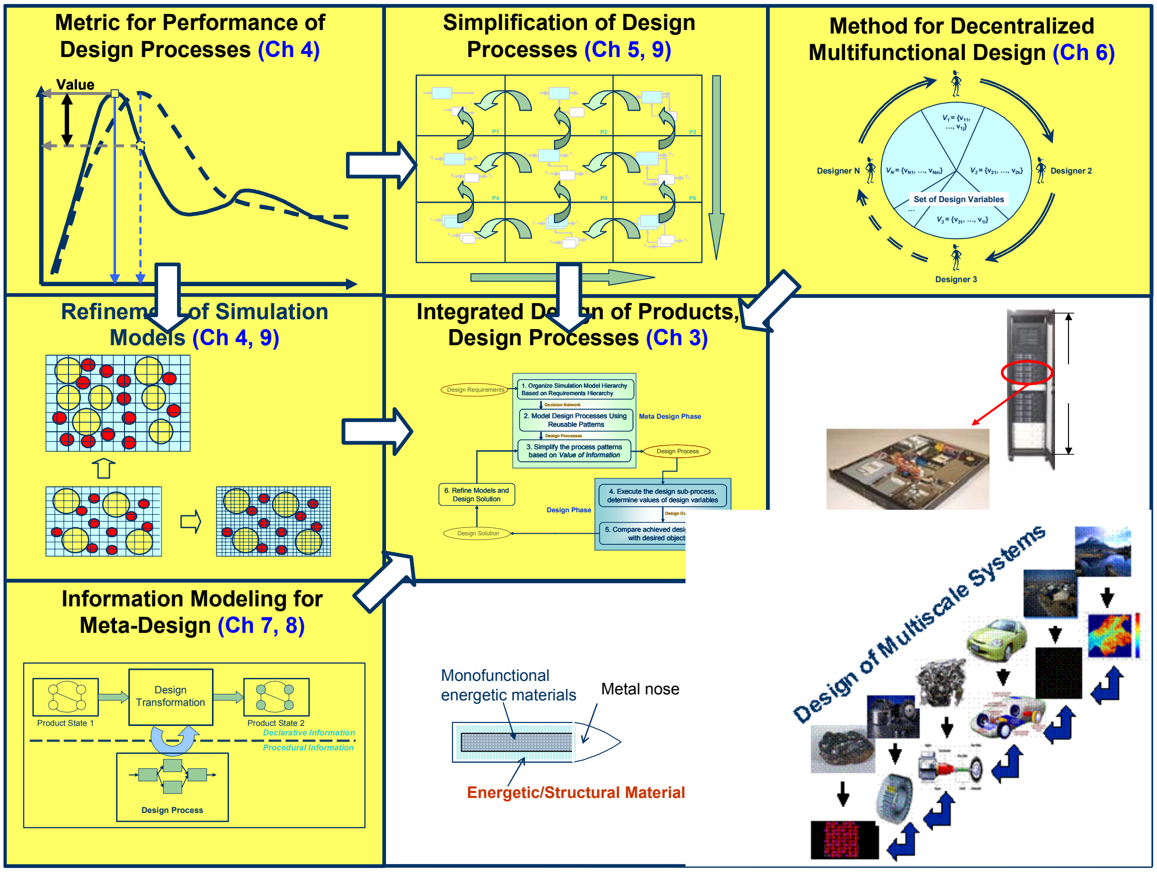


Figure 1-11 – Overview of research contributions presented in various chapters in the dissertation

The research contributions associated with each research question are summarized in Table 1-5 and are categorized into three headings – 1) robust multiscale design exploration method, 2) methods and metrics for design process simplification and model refinement, and 3) 3-P information modeling approach. The details of these three categories of research contributions are discussed next.

1. *Robust Multiscale Design Exploration Method (RMS-DEM):* As an answer to the first research question, a two phase, six-step method for designing products and design processes in an integrated fashion is developed. The method is discussed in detail in Chapter 3. The method is based on the principles for systems approach

for meta-design, robust design, and stepwise refinement of products and design processes.

2. *Design Process Refinement and Model Refinement:* One of the primary contributions from the answer to the second research question is the development of a Value of Information based metrics that can be used for making meta-design decisions. The metric is used in methods for design process simplification and simulation model refinement. Steps for scale decoupling and decision decoupling are developed and validated. The details are discussed in Chapter 4 and Chapter 5. A method for designing multifunctional systems when the functional aspects of the problem are controlled by decentralized teams is developed in Chapter 6. This method is based on propagating sets of design space.
3. *3-P information Modeling Approach:* The third set of research contributions is an approach for modeling design information to support meta-design. The approach is based on modularity of design information and separation of product, problem and process related information. Information models for product, problem, and process information are developed. A strategy for implementation of these concepts on existing design frameworks via separation of declarative and procedural information is presented and validated. The details of the approach are discussed in Chapter 7 and Chapter 8.

It is emphasized again that the focus of this dissertation is not on developing domain specific multiscale models, where mathematical modeling of physical phenomena is important. The focus here is on design aspects. It is assumed that simulation models for modeling different physical phenomena at different scales are available. We do not address *how* different physical phenomena are linked at different scales, but the *impact* of those links *on design decisions*.

1.3 Validation Strategy for this Dissertation

Since the focus in this dissertation is on the development of design methods, validation entirely based on logical induction/deduction is not possible. This is because development of design methodology is based on a combination of subjective statements and mathematical modeling. Pederson and co-authors (Pedersen, Emblemsvag et al. 2000) developed a systematic framework for validation of design methods, called the validation square, which was later refined by Seepersad and co-authors (Seepersad, Pedersen et al. 2005). The validation square is based on the relativistic validation, which is a semi-formal and conversational process involving a gradual process of building confidence in the usefulness of new knowledge with respect to a purpose. The method consists of four phases and six steps. The four phases include theoretical structural validity (TSV), empirical structural validity (ESV), empirical performance validity (EPV), and theoretical performance validity (TPV).

The phases and steps in the validation square are shown in Figure 1-12. Structural validation is a quantitative process consisting of three steps: (1) accepting the individual constructs constituting the method, (2) accepting the internal consistency of the way the constructs are put together in the method, and (3) accepting the appropriateness of example problems used to verify the performance of the method. Steps 1 and 2 constitute the *theoretical structural validity* and are carried out by critical evaluation of literature for individual constructs' validity, and a flow chart based approach for establishing the internal consistency of the overall method. Step 3 constitutes the *empirical structural* *validity* and is facilitated by documentation of different viewpoints illustrating that the example problems are suitable and adequate for quantitative validation.

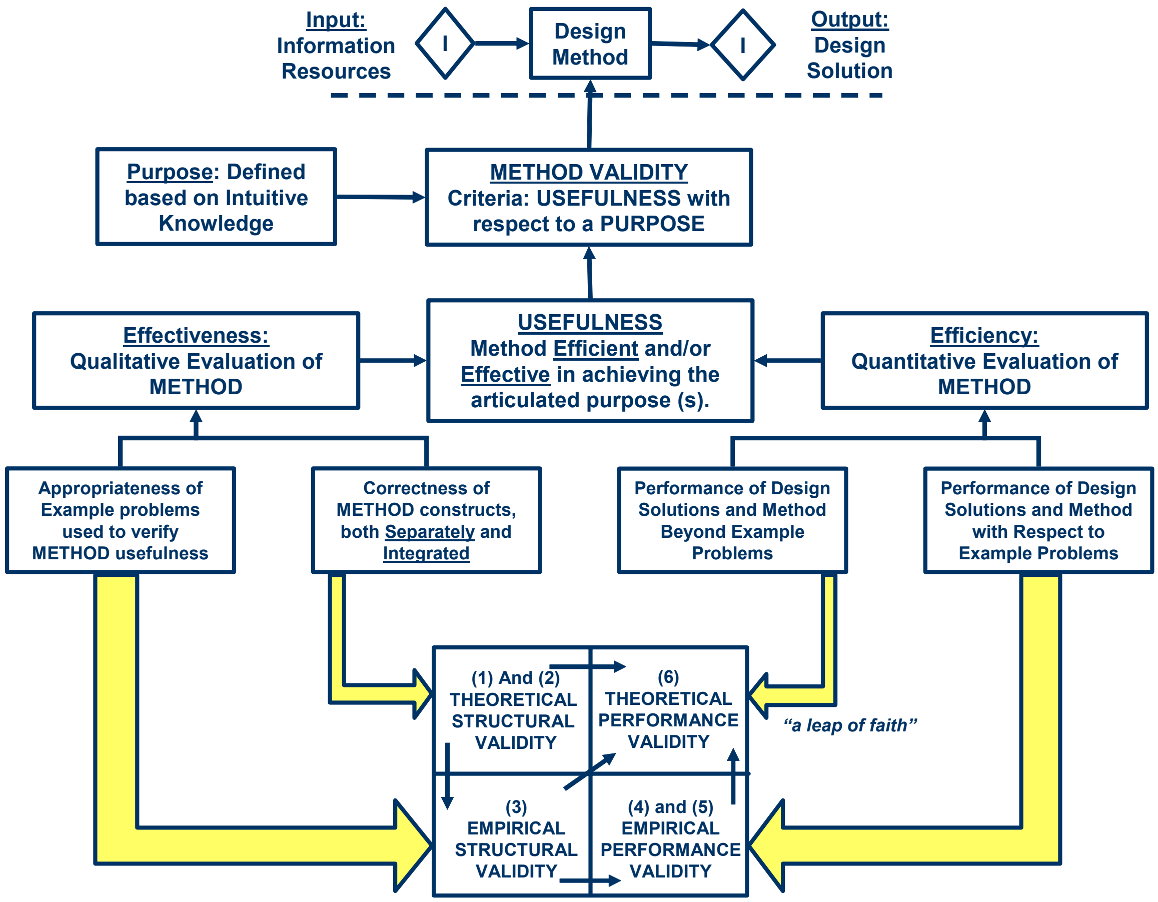


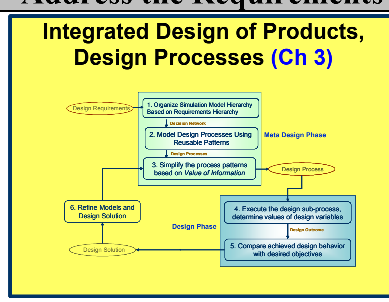
Figure 1-12 - Validation square (Seepersad, Pedersen et al. 2005)

Performance validation is a quantitative process consisting of the following three steps: (4) accepting that the outcome of the method is useful with respect to the initial purpose for some chosen example problems, (5) accepting that the achieved usefulness is linked to applying the method, and (6) accepting that the usefulness of the method is beyond the example problems. Steps 4 and 5 correspond to the *empirical performance validation* and are carried out by applying the methods to chosen examples and comparing the results with and without the use of the developed method. A quantitative evaluation is possible in this phase. Finally, Step 6 corresponds to theoretical performance validation, where arguments are presented to facilitate the leap of faith from few chosen examples to a broad set of scenarios. The purpose of the first five steps is to present enough evidence to facilitate this leap of faith.

In this dissertation, we use the validation square method for verification and validation of the design framework. Appropriate examples are used throughout this dissertation to demonstrate and validate the six components of framework for integrated design of products and design processes illustrated in Figure 1-11. An overview of the examples chosen to validate the components of the framework is provided in Table 1-6. Relevant portions of this table are referred to at the starting of each chapter to highlight the component of design framework developed or validated in that chapter.

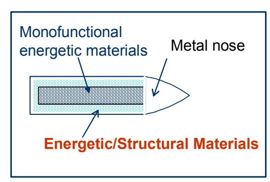
Table 1-6 – Components of the framework developed to address the requirements and examples used to validate the framework components

<table>
<tr><th>Framework Requirements</th><th>Components of the Framework Developed to Address the Requirements</th><th>Validation Examples</th></tr>
<tr><td>1) A method for integrated design of products and design processes</td><td>



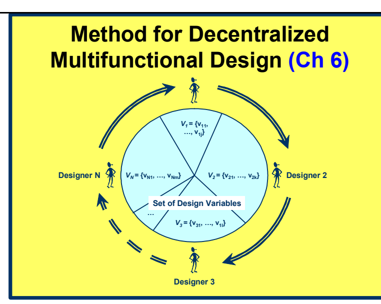
</td><td>

Materials-Product design example (Ch 9)



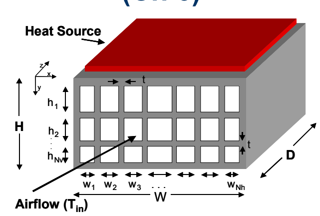
Purpose: To validate the method for integrated design of products and design processes

</td></tr>
<tr><td>2) Support for decentralized, multifunctional design</td><td>



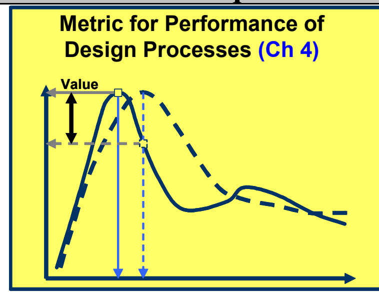
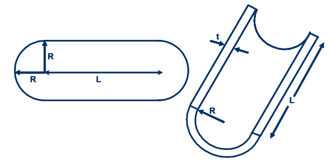
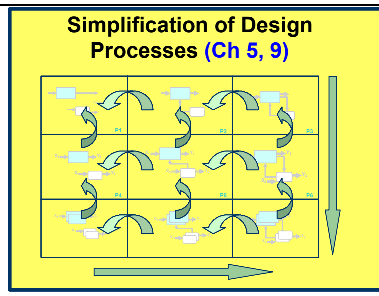
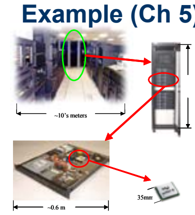
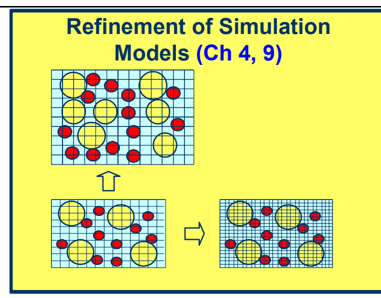
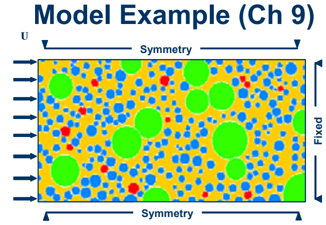
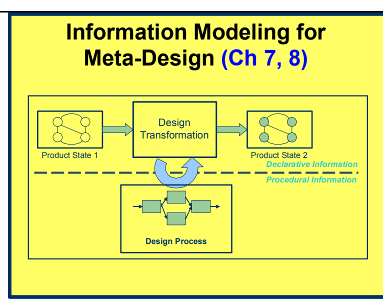
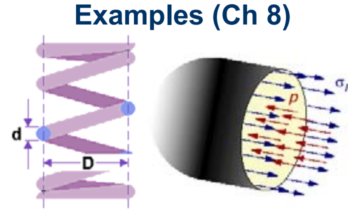
</td><td>

LCA Design Example (Ch 6)



Purpose: To validate the interval-based focalization method

</td></tr>
</table>

Framework Requirements	Components of the Framework Developed to Address the Requirements	Validation Examples
3) Metrics to quantify the performance of different design process alternatives		**Pressure Vessel Design Example (Ch 4)**  ***Purpose:*** To validate the value-of-information based metrics
4) Support simplification of complex design processes without affecting the performance of the product		**Datacenter Design Example (Ch 5)**  ***Purpose:*** To validate the use of value-of-information based metrics for design process simplification
5) Support evolving simulation models		**Particle Shock Simulation Model Example (Ch 9)**  ***Purpose:*** To validate the use of value-of-information based metrics for simulation model refinement
6) Support design process exploration, and reusability of existing design process, product and decision related information and knowledge		**Pressure Vessel, Spring Examples (Ch 8)**  ***Purpose:*** To demonstrate the approach for supporting meta-design in computational frameworks

The framework requirements, components developed to address that requirement, and the validation examples and their appropriateness are discussed in the following

1. A multiscale materials-product design example is used to validate the first component of the design framework – method for integrated design of products and design processes. This design example is chosen because it represents a general multiscale system where physical phenomena can be analyzed at multiple scales and both scales and physics are coupled with each other.
2. The second requirement of the framework is to support decentralized multifunctional design, for which, an interval-based method is proposed in Chapter 6. The method is validated using a Linear Cellular Alloy (LCA) design example. The example is chosen because of its multifunctional nature and is characterized by requirements from thermal and structural domains. The problem is also of reasonable complexity.
3. The third requirement is development of a metric for quantifying the performance of different design processes. The requirement is addressed in Chapter 4 via a new metric based on information economics. The metric is validated using a simple pressure vessel design example with a single variable. The example is chosen due to its simplicity in demonstrating the capabilities of the metric.
4. The forth requirement for the design framework is support for simplification of design processes, which is addressed in Chapter 5 via development of new methods for scale and decision decoupling. These methods are validated using a multiscale datacenter design scenario. The problem is chosen because of the

availability of models and design variables at different scales that permit scale and decision decoupling.

5. The fifth requirement is related to the support for evolving simulation models, which is addressed in Chapter 4 and validated in Chapter 9 using a particle shock simulation model. The particle shoch model is used for validating the refinement approach because of presence of multiple dimensions of refinement.
6. The sixth requirement for the design framework is computational support for meta-design, which is illustrated in Chapter 7, and validated using pressure vessel and spring design examples in Chapter 8.

A visual overview to aspects of validation square addressed in different chapters is provided in Figure 1-13.

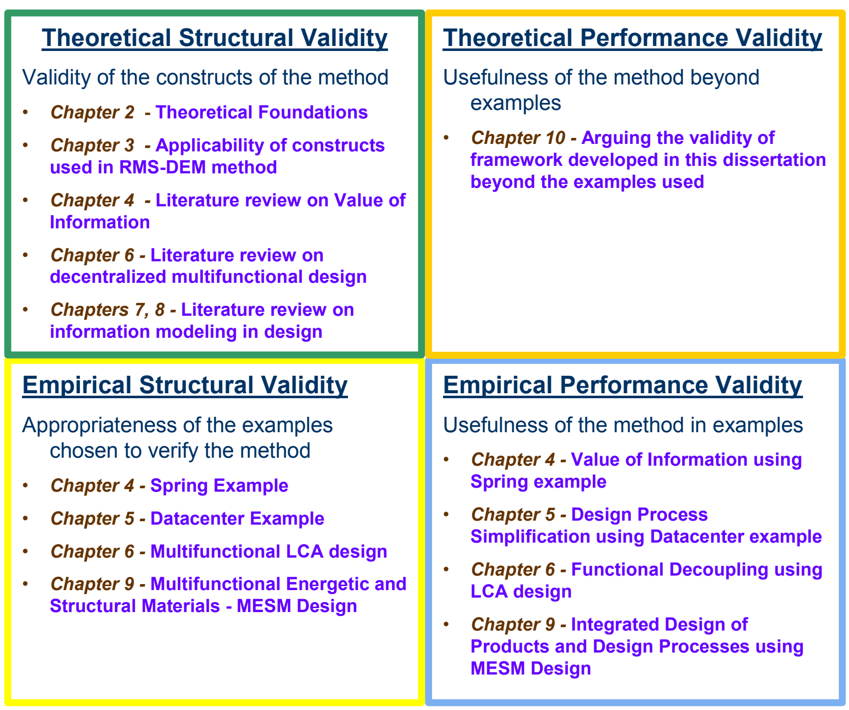


Figure 1-13 – An overview of validation strategy using the Validation Square (Pedersen, Emblemsvag et al. 2000)

The details of validation performed in each chapter are provided at the end of various chapters where examples are discussed (see Figure 4-30, Figure 5-34, Figure 6-15, and

Figure 9-48). Each of the hypotheses used to support the research questions is validated in Section 10.2. The overall validation squae for this dissertation is divided into five sub-squares for the five hypotheses. These validation sub-squares are labeled VSQ 1.1, VSQ 1.2, VSQ 2.1, VSQ 2.2, and VSQ 3.1; and are presented in Figure 10-2, Figure 10-3, Figure 10-4, Figure 10-5, and Figure 10-6. The relationship of overall validation strategy for the dissertation and the validation sub-squares is shown in Figure 1-10. Sectionwise details of the validation strategy for this dissertation are provided in Table 1-7.

An overview of the dissertation is provided in Figure 1-14 and discussed next.

In Chapter 1, the context and scope of this research are presented. The challenges associated with modeling and designing multiscale systems are presented. These challenges provide the motivation and frame of reference for designing design processes in conjunction with the products. The research questions and associated hypotheses are discussed. The expected contributions are summarized, and a validation strategy for the dissertation is established.

In Chapter 2, the theoretical foundations for designing simulation-based design processes are discussed. These foundations include existing design constructs such as decision-based design, meta-design and Decision Support Problem Technique, robust design, utility theory, compromise Decision Support Problem, information economics, interval arithmetic, and information modeling. Relevant literature for each of these areas is referenced, discussed, and critically evaluated to show the appropriateness of use of these constructs for the design framework developed in the dissertation. The literature review in Chapter 2 is used to identify availability, strengths, and limitations of these constructs in the context of integrated design of products and design processes, and becomes an essential component of theoretical structural validation.

Table 1-7 – Validation strategy for this dissertation

Theoretical Structural Validation		
	• Critical review of the literature foundational to complex multiscale systems design and designing design processes. Topics include multiscale modeling, DSP Technique, multidisciplinary analysis and optimization • Discussion of advantages and limitations of available approaches and identifying research opportunities	Sections 2.2, 2.3, 2.4, 2.5
	• Critical review of literature related to Design Structure Method (DSM) for designing design processes	Sections 3.3.1, 3.3.2
	• Critical review of literature on value of information metric	Section 4.2
	• Literature review on decentralized multifunctional design	Sections 6.2, 6.3
	• Critical review of literature on modeling design processes	Section 2.6
	• Presentation of the design method for integrated design of products and design processes. Presentation of consistency and the appropriateness of constructs in the design method	Section 3.5
Empirical Structural Validation		
	• Discuss the appropriateness of *structure design* problem to demonstrate the use of design method	Section 3.4
	• Discuss the appropriateness of *pressure vessel* design problem to show the use of value of information metric	Section 4.5
	• Discuss the appropriateness of *multiscale datacenter cooling system design* example for design process simplification through decoupling	Section 5.5
	• Discuss the appropriateness of *Linear Cellular Alloy* design example for decentralized multifunctional design	Section 6.5
	• Discuss the appropriateness of *multiscale materials* design example for validating the design method	Section 9.7
Empirical Performance Validation		
	Use the examples to demonstrate the utility of design framework by answering the following questions: • Does the method for integrated design of products and design processes improve efficiency and effectiveness of utilization of information? • Is it useful to define design processes in terms of reusable interaction patterns? • Is the value of information metric suitable for making meta-level decisions such as determining the right level of *a)* simplification of design processes, and *b)* refinement of simulation models? • Is value of information guided decoupling of scales and decisions helpful in increasing the efficiency of decision making without affecting the designers' decision making capability • Is the method proposed for decentralized decision making more effective than traditional methods? • Is the information modeling strategy involving separation of product, process and problem related information useful for information reuse?	Sections 3.5, 9.7 Section 3.5.2 Section 4.5.3 Section 9.7 Section 5.5 Section 6.5 Section 8.4
	Demonstrate that the observed usefulness is linked to the constructs developed in this dissertation.	Chapters 4, 5, 6, 9
	Verify the claims using numerical results obtained from the example scenarios	Chapters 4, 5, 6, 9
Theoretical Performance Validation		
	Build confidence in the generality and usefulness of the approach beyond the specific example problems. Argue that the approach is useful for the example problems and that the example problems are representative of general problems	Section 10.2.4

In Chapter 3, the details of the steps in the proposed design method are discussed. The key elements of the design method are discussed from the perspective of embodying Hypotheses H1.1 and H1.2. Steps in the design method are illustrated using a structural design problem. Verification of internal consistency of the method is emphasized in this chapter for theoretical structural validation. Advantages and limitations of the method are discussed in the context of simulation-based design of products and design processes.

In Chapter 4, an information economics based metric for determining the value of information is developed for quantifying the impact of design process and comparing different process options. This metric is used in two steps in the design method. Theoretical structural validation of the metric developed in that chapter is performed by a critical evaluation of literature and identifying the need for a new metric. Empirical structural and empirical performance validity of the metric is carried out in that chapter by showing the appropriateness of pressure vessel design example and showing that the quantitative results from the example demonstrate the usefulness of the metric.

In Chapter 5, a method for using value of information metric in design process simplification is developed and validated. The method embodies hypotheses H2.1 and H2.2. The focus of simplification in the dissertation is limited model and decision decoupling. The method is validated using a datacenter cooling system design example. Empirical structural is performed by showing that a datacenter cooling system represents a multiscale system with different models for predicting the behavior at different scales. Empirical performance validity of the method for simplification is carried out by showing that the results from the datacenter design example are inline with the claims in the design method.

In Chapter 6, functional decoupling of multifunctional design problems is addressed. A method for decoupling weakly coupled systems using value of information is discussed. In addition to that, a method for design of multifunctional systems in a decentralized environment is also presented. The method is based on passing ranged sets of specifications between designers in-charge of different functional requirements in a cyclic manner. The method is validated using a Linear Cellular Alloy design scenario. Both empirical structural and empirical performance validation are performed in that chapter.

In Chapter 7, an information modeling strategy for modeling design processes to support the integrated design of products and design processes is presented. The strategy is based on separating product, problem, and process related information. The strategy is an embodiment of hypothesis H3.1. The validation focus in that chapter is on theoretical structural validation. Existing literature on design process modeling is critically evaluated and the requirements for a new design information modeling strategy are presented.

In Chapter 8, an implementation of the information modeling strategy presented in Chapter 7 is provided. The chapter serves to validate the constructs for modeling design information developed in the dissertation. Empirical structural and empirical performance validations are performed using two examples problems – design of pressure vessel and design of spring.

In Chapter 9, the design methods developed in the dissertation are applied to a multiscale, multifunctional materials design problem. The problem is to design a material with multifunctional energetic and structural properties that can be used in a projectile to replace a portion of the structural material. The problem is modeled as an integrated design of materials, products, and design processes. In addition to the validation of design methods, the chapter is also crucial from the standpoint of materials design domain. In this chapter, we discuss the validation of the proposed design methods and value of information metrics.

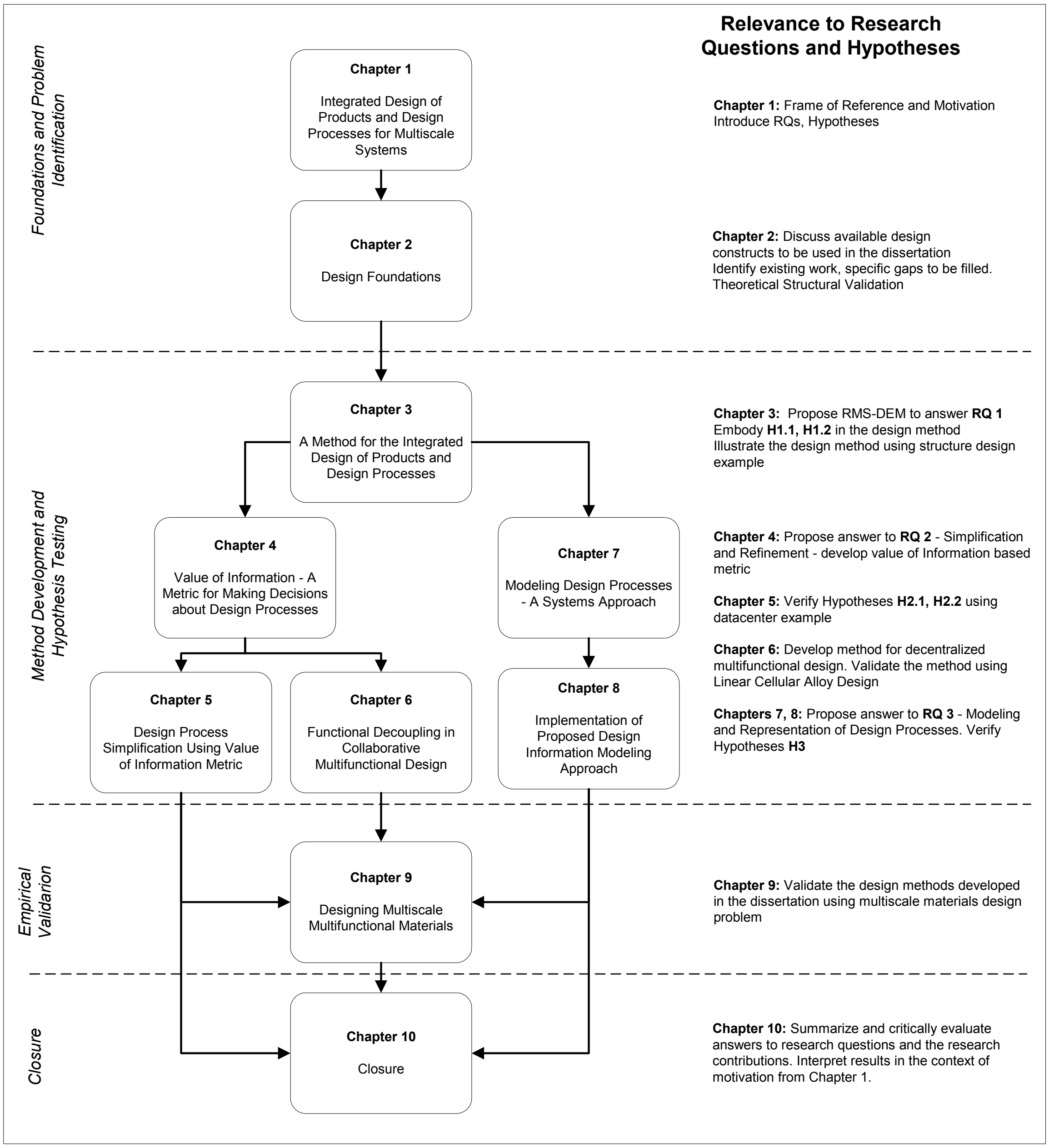


Figure 1-14 - Dissertation overview and roadmap

In Chapter 10, the dissertation is summarized and the intellectual contributions are critically reviewed. The advantages and limitations of the methods, metrics, and information modeling strategy are discussed. For theoretical structural validation, it is argued that these constructs are valid beyond the example problems selected for empirical validation. Finally, avenues for future research and broader applications of the fundamental ideas in this dissertation are discussed.

Chapter 2 Foundations for the Framework for Integrated Design of Products and Design Processes

In this Chapter, we present the foundational concepts used in the dissertation for the framework for integrated design of products and design processes. We start with providing a relationship between the designing design process effort in this dissertation and other research efforts in the design community. This relationship is shown in Section 2.1. In the remaining part of the chapter, we present constructs from existing research efforts in design that are foundational to the framework for integrated design of products and design processes. These include a*)* decision-based design and a specific instantiation - Decision Support Problem Technique, *b)* robust design, *c)* utility theory, *d)* information economics, and *e)* design information modeling. An overview of the aspects of these constructs discussed in this chapter and their role throughout the dissertation is presented in Table 2-1. A brief discussion of the table follows.

a) *Decision-based design* is the perspective from which the design methods and metrics are developed. Decision Support Problem (DSP) Technique is an instantiation of decision-based design. Constructs such as Compromise DSP, Selection DSP developed as a part of the DSP Technique literature are used in the design method developed in this dissertation. Decision-based design and DSP Technique are discussed along with its relevance in the dissertation in Section 2.2. Decision based design is used in this dissertation as a philosophy on which the design of products and design processes is based, which is discussed in Section 3.2. The DSP Technique is used for developing the strategy for design information modeling via a decision problem based approach, which is discussed in Section 7.2.

Table 2-1 – Role of constructs discussed in Chapter 2 throughout the dissertation

Section #	*Construct*	Purpose of Discussion in Chapter 2	Use in the Dissertation
Section 2.1	*Designing Design Processes in Conjunction with Products*	- Overview of existing design methodologies (Section 2.1.1) - Review of literature on design methods - Review of literature on designing design processes (Sections 2.1.2 and 2.1.3)	- Theoretical Structural Validation for design method presented in Section 3.5
Section 2.2	*Decision-Based Design and DSP Technique*	- Overview of decision-based design and specific instantiation (DSP Technique) - Overview of decision-based modeling of design processes	- Philosophy for design and meta-design (Section 3.2) - Strategy for modeling design processes via separation of decision problem, product, and process information (Section 7.2)
Section 2.3	*Robust Design*	- Overview of robust design - Literature review on utilization of robust design for decision making under uncertainty	- Decision making in the presence of uncertainty (Section 3.5.4) - Decision making in the presence of process simplification (Section 5.2.2)
Section 2.4	*Utility Theory*	- Overview of utility theory and means for modeling designers' preferences	- Modeling designers' preferences (Sections 4.4, 5.3.2, 5.4.2, and 9.3) - Development of Value-Of-Information metric (Section 4.3)
Section 2.5	*Information Economics*	- Overview of information economics - Literature review of available metrics for value of information	- Development of Value-Of-Information metric (Section 4.3)
Section 2.6	*Design Information Modeling*	- Overview of available information modeling strategies - Literature review on design information modeling - Identification of research gaps to support meta-design	- Establishment of 3-P information modeling strategy (Section 7.3) - Development of information models for Products, Processes, and Decision Problems (Sections 8.1, 8.2, and 8.3)

b) *Robust design* refers to the class of design methods developed to design systems that are insensitive to variations in the environment, while achieving the desired specifications as closely as possible. Robust design concepts are used in this dissertation for making decisions in the presence of uncertainty (see Sections 3.5.4 and 5.2.2). Details of robust design and its role in this dissertation are discussed in Section 2.3.

c) *Utility theory* is used to model rational decision making in a mathematically rigorous form. In this dissertation, utility functions are used to model designers' preferences for making decisions related to products and design processes (see Sections 4.4, 5.3.2, 5.4.2, and 9.3). Utility theory is also used for developing the value of information metric in Section 4.3. The details of utility theory and its use in the dissertation are discussed in Section 2.4.

d) *Information economics* refers to the field where impact of additional information on the quality of decisions is studied. A component of the research in information economics involves development of metrics that quantify this impact. The concepts from information economics are used in this dissertation to make design-process related decisions based on the value of information metric (see Section 4.3). An overview of foundational work in the field of information economics which is relevant to this dissertation is provided in Section 2.5.

e) Finally, in Section 2.6, an overview of the literature on *design information modeling* is presented. The focus is on providing an overview of available information modeling strategies, a literature review on design information modeling, specifically on models for representation of product and process specific information. Based on this literature review, the research gaps for support meta-design are identified. The concepts from design information modeling (Section 2.6) are used to establish 3-P information modeling strategy in Section 7.3 and develop information models for Products, Decision Problems, and Processes in Sections 8.1, 8.2, and 8.3 respectively.

2.1 Frame of Reference – Designing Design Processes in Conjunction with Products

2.1.1 Designing – A Goal Oriented Activity

While natural sciences are concerned with how things are, an engineer, and more generally a designer is concerned with how things ought to be in order to attain goals and to function (Simon 1996). As pointed out by Braha and Maimon (Braha and Maimon 1997), the distinction between engineering science and natural science is that the aims and methodology of engineering science differ, i.e., natural sciences are concerned with analysis and engineering with synthesis; natural science is theory oriented while engineering is result oriented. This distinction between design activities and natural science is embodied by many researchers in their definitions of 'design'. Suh defines design as interplay between what we want to achieve and how we want to achieve it (Suh 1990). Mistree and coauthors view design as the conversion of information that characterizes the needs and requirements for a product into knowledge about the product (Mistree, Smith et al. 1990). The metagoal of design is to transform requirements – generally termed functions, which embody the expectations of the purposes of the resulting artifact, into design descriptions (Gero 1990). The National Science Foundation defines design as the process by which products, processes and systems are created to perform desired functions through specification.

Models for Design

In order to support the development of design, the research in engineering design is categorized into design philosophies, models, and methods. Design theory is a collection of principles that are useful for explaining a design process and provide a foundation for basic understanding required to propose useful methodologies. Design theory explains what design is, whereas design methodology is a collection of procedures, tools and techniques for designers to use when designing. Design methodology is prescriptive, while design theory is descriptive (Finger and Dixon 1989; Finger and Dixon 1989; Evbuomwan, Sivaloganathan et al. 1996). Design methods have been developed from different viewpoints that emphasize different facets of the overall design process. Some of these views as summarized by Evbuomwan and coauthors include *a)* design as a top-down and bottom-up process, *b)* design as an incremental (evolutionary) activity, *c)* design as an knowledge-based exploratory activity, *d)* design as an investigative (research) process, *e)* design as a creative (art) process, *f)* design as a rational process, *g)* design as a decision-making process, *h)* design as an iterative process, and *i)* design as an interactive process. Although design methods are generally developed with a few of these viewpoints in mind, an ideal design method should support all of these.

Pahl and Beitz (Pahl and Beitz 1996) identify four key phases that are common to any prescriptive model for design. These phases include planning and clarification of task, conceptual design, embodiment design, and detail design. Planning and clarification of task involves identifying the requirements that the outcome of design should fulfill. These requirements are then converted into a statement of the problem to be solved. Conceptual design involves generation of principles used to satisfy the problem statement. Embodiment design involves refinement of the solution for the purpose of eliminating those that are least satisfactory until the final solution remains. During the detail design, all the details of the final design are specified and manufacturing drawings and documentation are produced.

In contrast to the descriptive models of design, prescriptive models exemplify how design should be done and not necessarily how it is done. Most of the prescriptive methods of design are based on the assumption that any design activity consists of three core activities – analysis, synthesis, and evaluation. *Analysis* is defined as the resolution of anything complex into its elements and the study of these elements and of their relationships. *Synthesis* is putting together of parts or elements to produce new effects and to demonstrate that these parts create an order (Pahl and Beitz 1996). A general model of design can be visualized as a feedback loop of synthesis, analysis and evaluation. It is important to note that although the focus in natural sciences and design is on analysis and synthesis respectively; both analysis and synthesis form the key components of any design activity. These general ideas for analysis, synthesis, and evaluation are described by Gero (Gero 1990) as a series of transformations of information starting with the requirements and ending with a description of the design that satisfies the requirements. According to Gero, the key aspects of product information include function (F), structure (S), expected behavior (Be), achieved behavior (Bs), and product description (D) (see Figure 2-1). Function (F) is the relation between the goal of a human designer and the behavior of the system. Structure (S) represents the artifact's elements and its relationships. The structure is also called the form of the artifact. The achieved behavior (Bs) of the structure is directly derivable from its structure using the laws of physics. The expected behavior (Be) represents the physical properties that the artifact should have in order to satisfy the functional requirements (F). The product structure can be converted in a manufacturable product description (D). In terms of these definitions, analysis is the transformation of product structure to achieved behavior and synthesis is the transformation from expected behavior to structure. Evaluation refers to comparison of the expected behavior with achieved behavior. This is an iterative process. The ASE view of design is foundational to many design efforts such as Shimomura (Shimomura, Yoshioka et al. 1998) and Maher (Maher 1990).

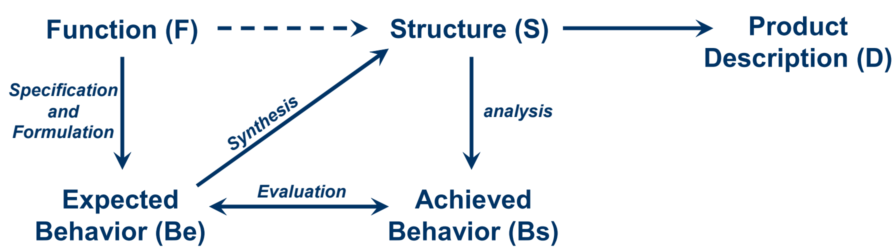


Figure 2-1 – Model of design as a process (Gero 1990)

Design of Systems

The general idea of Analysis, Synthesis, and Evaluation is extended to the domain of systems engineering where the focus is not only on the design of simple products for a given set of requirements, but on complex systems throughout their lifecycle. A system is defined as a group of associated entities which is characterized by a mental construct; one of the associated entities is the boundary (Muster and Mistree 1988). A system is also defined as a set of components (subsystems, segments) acting together to achieve a set of common objectives via the achievement of a set of tasks. Systems theory as an inter-disciplinary science uses special methods, procedures, and aids for the analysis, planning, selection and optimization of complex systems (Pahl and Beitz 1996). The systems approach to design reflects the general appreciation that complex systems are best tackled in fixed steps, each involving analysis, synthesis, and evaluation (Pahl and Beitz 1996). Hence, these steps are carried out during all the phases of the lifecycle including planning, preliminary study, system development, system production, system installation, system operation, and replacement. In a systems theory process model, the steps of analysis, synthesis, and evaluation repeat themselves in so called lifecycle phases of the system in which the chronological progression of a system goes from abstract to concrete.

A design process defines what the system must do, how well the system must do it, and how the system should be tested to verify and validate the system's performance (Buede 2000). A commonly adopted process for systems design is based on the Systems Engineering Vee (Buede 2000), which is based on successive decomposition of systems into subsystems followed by recomposition. During the decomposition process, the originating requirements are analyzed and defined in engineering terms and then partitioned into a set of specifications for several segments, elements and components (Buede 2000). The focus of decomposition process is the movement from need to system level requirements to specifications for each subsystem to the specification of each component. The focus of recomposition is the integration of individual components to subsystems and subsystems into larger systems. The integration involves testing of the newly assembled system and determining whether the assembled system satisfies the system requirements. Verification and validation of the system are also carried out during the integration phase.

Simulation-Based Design

With the development of computers, the design activities are increasingly carried out by using their computational capabilities. Over the last few decades, the role of simulation in engineering design has evolved from failure analysis and design verification to parametric design optimization. Until recently, the simulation models were used only in the detailed design phases, where most of the design information is available. Further, the simulation models were developed for analyzing the behavior of individual components in an independent fashion. With the increase in computational capabilities and the development of methodology for composing component simulation models together to develop overall system simulations, it is now progressively possible to evaluate the emergent behavior of complete systems. These capabilities have elevated the role of simulation in design from mere component failure analysis and parametric optimization to systems design and given rise to the field of simulation-based design. The general design methods for design using simulations remains the same – the key activities being analysis, synthesis, and evaluation. The only difference in simulation-based design and conventional design is that computer simulations are used to support the three activities – analysis, synthesis, and evaluation.

2.1.2 Designing Design Processes

Design processes represent the tasks carried out from requirements to the final design, their ordering, and the information flow between them. The design processes are currently structured based on the design method used. The mapping of product's functional characteristics to concepts, and concepts to physical parts primarily dictates the architecture of design processes. In systems design, the product's functional decomposition determines the analysis and synthesis tasks performed. The hierarchy of functions determines the sequence in which the tasks need to be carried out. Generally, the dependencies between product's functions dictate the dependencies between information flows between tasks in the design process. For complex systems, the requirements of the entire system cannot be transformed into a single stage to detailed specifications. The transformations are carried out in several successive stages. Given the requirements for a complex system, the analyses are performed to transform these requirements into system specifications. These specifications flow down to the next level as subsystem requirements; the subsystems must be designed to achieve these system level specifications. The subsystem level specifications then become requirements for lower level subsystems until the lowest level details are decided on (Koch 1997). Such a partitioning of requirements into subsystem requirements and the associated requirements flow down defines the design processes for such complex systems. As mentioned by Koch (Koch 1997), unlike the pre-existing hierarchies in organizations, the hierarchies in products are created through a process of formal decomposition or informal partitioning. Hence, the design processes in hierarchical systems are dependent on the kind of partitioning chosen for products.

Simon (Simon 1996) pointed out that the design process strategies can affect not only the efficiency with which the resources are used for designing, but also the nature of final design as well. He further asserted that both the shape of the design (artifact) and the shape and organization of the design process are essential components of the theory of design (Simon). Mistree and co-authors describe the role of designing design processes as follows: "Compared to the traditional engineering design in which synthesis of the product plays the central role, the synthesis of the process (which includes design, manufacture, and support aspects) is the dominant feature in concurrent engineering. With the synthesis of the process at this higher level, the synthesis of product follows naturally" (Mistree, Smith et al. 1990). In spite of the importance of design processes, the design of design processes is still carried out in an adhoc manner.

2.1.3 Current Trends in Improving the Design Process – Increasing Concurrency and Simultaneous Consideration of Different Aspects

Recently, efforts towards improving the design processes are focused on *a)* performing activities in a concurrent fashion so that maximum possible amount of information is available for designing, and *b)* including complete lifecycle considerations upfront during the design phase. Researchers have shown benefits of including lifecycle considerations in terms of reduction in costly design iterations. Approaches to design incorporating lifecycle considerations include concurrent engineering (Pennell and Slusarczuk 1989; Kusiak and Park 1990; Bowen and Bahler 1992; Bras 1992; Mistree, Smith et al. 1993; Bahler, Dupont et al. 1994; Prasad 1996; Cantamessa and Villa 2000), simultaneous engineering, United Life Cycle Engineering, producibility engineering, Integrated Product and Process Design (IPPD), and Product Lifecycle Management.

Concurrent Engineering: Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support. The approach is intended to cause the developers, from the outset, to consider all elements of the product lifecycle from conception through disposal, including quality, cost, schedule, and user requirements (Winner, Pennell et al. 1988). According to Beude (Buede 2000), the term concurrent engineering simply means that the systems engineering process should be done with all the phases of the lifecycle in mind. Simultaneous engineering, ULCE, and producibility engineering are efforts similar to the concurrent engineering.

Integrated Product and Process Design: One of the specific embodiments of the ideas from concurrent engineering is Integrated Product and Process Design (IPPD). IPPD indicates, in the broadest sense, the overlapping, interacting, and iterative nature of all of the aspects of the product realization process. The method is a continuous process whereby a product's cost, performance and features, value, and time-to-market lead to a company's increased profitability and market share (Magrab 1997). In the IPPD effort, the focus is mainly in integrating product design with the manufacturing process. The strategy adopted for IPPD is to include manufacturing knowledge as much as possible in the design decision making.

Product Lifecycle Management: Recently, the ideas from concurrent engineering and IPPD are extended to a broader umbrella of Product Lifecycle Management, which is defined by IBM as "…a strategic approach to creating and managing a company's product-related intellectual capital, from its initial conception to retirement" (IBM 2004). Accordingly, "PLM improves a company's product development processes and its ability to use product-related information to make better business decisions and deliver greater value to customers". JD Edwards (Edwards 2002) defines PLM as "management of a series of business processes, enabled by collaborative applications that manage a portfolio of products … to maximize market share and profitability". CIMData defines PLM as "a strategic business approach that applies a consistent set of business solutions in support of the collaborative creation, management, dissemination, and use of product definition information across the extended enterprise from concept to end of life – integrating people, processes, business systems, and information". Generally, PLM is taken to be a strategic business approach for the effective management and use of corporate intellectual capital (Fenves, Sriram et al. 2003).

Product Lifecycle Management (PLM) involves activities from the initial conception to retirement of the product and is aimed at improving the product development process. The goal in PLM is to integrate all the product realization activities including market planning, concept development, design, production, sales, marketing, etc. Considering the field's extensive scope there are numerous interpretations, each highlighting different facets of import. Examples include a) interoperability issues and standardization in CAD/CAM/CAE, b) overarching management considerations c) collaboration d) information management and sharing, and e) integration.

One of the key components of PLM with regard to this aspect is the integration of the process with the design of the product. Although design processes play a crucial role in PLM, integrating the design of "design processes" with the product has received little attention. *Systematic methods for designing design processes* have not been formalized. Additionally, while it is true that the potential of leveraging components of existing products towards developing new products has been exploited, the possibility of leveraging sub-processes in new product realization scenarios is substantial.

2.1.4 Designing Processes in this Dissertation

Although these efforts are focused on *improving* the design processes for efficient and effective design by taking concurrency into account, we believe that a focused effort at *designing* the design processes in association with the products is required. This is especially true in the complex systems where the design processes cannot be developed just based on functional decomposition. One such special class of complex systems that can benefit from designing design processes is ‘Multiscale Systems’. For multiscale systems such as materials, it is not possible to decompose the requirements in terms of functions that have concepts associated with them. Hence, development of design processes based on functional decomposition is not an option. There is a need for designing a design process that is suitable for the design requirements at hand. Further, different design processes are suitable for designing the same multiscale system as the requirements change. The challenges associated with designing multiscale systems necessitate development of new design methods that are based on the consideration of both design process and the product. This need for developing a new design method for considering design of product and design process is addressed in this dissertation.

2.2 Decision-Based Design and the DSP Technique

“To be successful, the engineering design of systems must embrace the notion that many decisions are made during the development process. This is not a controversial position to take. However, adopting the notion that these decisions should be made via a rational explicit process is not consistent with much of the current practice in the engineering of systems.” (Buede 2000). It is this formalization of rational decision making process that the decision-based design is focused on.

According to many researchers such as Hazelrigg (Hazelrigg 1998), Muster and Mistree (Muster and Mistree 1988), and Thurston (Thurston 1999) the fundamental premise of decision-based design is that engineering design is primarily a decision-making process. Decision-based design (DBD) emphasizes a perspective from which design methods can be developed (Mistree, Smith et al. 1990). From a decision-based design perspective, the principal role of a designer is to make decisions. In decision-based design, decisions serve as markers to identify the progression of a design from initiation to completion. The paradigm of decision-based design is based on decisions made by designers as opposed to design that is assisted by the use of computers as well as optimization methods, or methods that evolve from specific analysis tools such as finite element analysis. Decisions help bridge the gap between idea and reality, and are a unit of communications that are characterized by information from many sources and disciplines and may have both discipline-dependent and discipline independent features. In DBD, the making of decisions causes the transformation of information into knowledge. Accordingly, a design process from the DBD perspective is conversion of information that characterizes the needs and requirements for a product into knowledge about the product (Mistree, Smith et al. 1990; Bras and Mistree 1991). The role of decisions in design processes is also pointed out by Gero (Gero 1990) in his statement – "a prevalent and pervasive view of designing is that it can be modeled using variables and decisions made about what values should be taken by these variables."

The implementation of DBD can take many forms. One of the implementation approaches is the Decision Support Problem Technique. The Decision Support Problem Technique (Muster and Mistree 1988) provides support and rationale for using human judgment in design synthesis. According to this technique, designers are and will be continued in two primary activities – *processing symbols* and *making decisions,* independent of the approaches or methods used to plan, establish goals and model systems. Hence, the assertion is that the process of design, in its most basic sense is a network of decisions. As the engineering problems increase in complexity and the interaction of systems with their environments become more and more unpredictable, the engineers need an approach to negotiating solutions to their problems that permits designers to accept a satisfying solution instead of vying for optimal solutions. The DSP Technique helps in partitioning the problems in simple terms so that it is possible to find solutions for it, while being close to the actual system.

DSP Technique for increasing efficiency and effectiveness

According to Mistree and coauthors (Mistree, Smith et al. 1990), the efficiency and effectiveness of designers are increased by increasing the speed with which the design iterations are completed and by reducing the number of iterations. The increase in efficiency via increase in the speed of iteration has been the focus of design automation. *To achieve the reduction in the number of iterations, there is a need for a model of the process and information for determining how the process can be improved*. Designers can use the DSP Technique to increase the efficiency and effectiveness of design process by reducing the number of iterations and increasing the speed of iterations. The speed of design iterations can be increased if the parts of the design process can be modeled on a computer, and reducing corrective design can reduce the number of iterations. Hence, the necessary ingredient in increasing efficiency and effectiveness of human designers is modeling of design process in a manner that they can be analyzed, manipulated and implemented. By focusing on decisions, the DSP Technique provides a means for modeling design processes in a manner that can be analyzed and reconfigured.

Meta-design and design phases in the DSP Technique

Decisions are based on the human preferences and involve allocation of resources. Decisions made during a design process are of two types: decisions about the product and decisions about the process through which the product is designed. Some examples of decisions about the product include selection among alternative concepts, determining values for design variables, etc. Examples of process decisions include the sequence in which design decisions will be made, the manner in which different activities will be performed, the way tasks will be executed, etc. Similar to the decisions about product, decisions made about design processes also play a significant role in the way design is carried out. Hence, the ability to structure design processes in the most effective manner it important.

The DSP Technique consists of two phases – *meta-design* phase and *design* phase. The meta-design phase consists of planning and structuring of support problems and the design phase consists of solving the support problems and post solution analysis (see Figure 2-5). *Meta-Design* is a meta-level process of designing systems that includes partitioning the system for function, partitioning the design processes into a set of decisions and planning the sequence in which these decisions will be made. In this phase, product specific decisions themselves are not made or even pursued, but rather, the design process to be implemented in design phase is itself designed. The designer specifies a process in terms of certain base entities (e.g., phases, events, decisions, tasks, etc.). All information is considered as relationships between inputs and outputs. These entities are used to build directed networks modeling design processes that can be manipulated. In the actual *design* phase, the solution to the design process is sought and validated. An important aspect in implementing the DSP Technique is to create software tools and systems to help designers to concurrently create models of design processes and DSPs and foe group of designers for various fields of engineering to work at the same design project at the same time.

Modeling design processes using DSP Technique palette

The definition of design used by the authors in (Muster and Mistree 1988) is:

"*Designing is a process of converting information that characterizes the needs and requirements for a product into knowledge about the product*."

In the context of this definition, the conversion of information into knowledge invariably takes place throughout the design timeline. Throughout the timeline, the types of decisions being made are the same in all stages and the quantity of hard information increases as the knowledge about the product increases. The assertion in the DSP Technique is that any process in design can be defined in terms of phases and events.

In the DSP Technique, the entities for designing a design process are contained in the DSPT Palette. These entities can be used to model design process hierarchically and in a domain independent fashion. The DSPT Palette contains three different classes of entities – potential support problem entities, base entities and transmission entities. The potential support problem entities are phases, events, tasks, decisions and systems (see Figure 2-2). Icon "P" denotes a phase. The phases are used to represent elements of partitioned process. For example, conceptual design, embodiment design etc. are phases in the Pahl and Beitz design method. Events are denoted be "E". Events occur within a phase. Examples of events are – "evaluate for economic criteria", "check for system feasibility" etc. Phases and events are accomplished by tasks and decisions. Tasks and decisions require direct involvement of human designers. A task is any activity to be accomplished. In DSP Technique, decisions are broadly categorized into selection decisions and compromise decisions. Selection decisions involve making a choice between a number of available options and compromise involves selecting the best possible values of design variables such that the system is feasible. A system can be either abstract or concrete and can be modeled by a grouping of associated subsystems.

Base entities are the most elementary entities for modeling design processes and can be implemented on a computer. The base entities are shown in Figure 2-2. The base entities are used to describe constraints and bounds on the design space, relationship between design variables etc.

Transmission entities in the DSPT palette are used to define the connections between various other entities used to model the design process. The transmission entities are based on the Miller's Living Systems Theory. The transmission entities include three types of basic transmissions – mass, energy and information and combinations of these three basic types. The transmission entities in DSPT palette are shown in Figure 2-2.

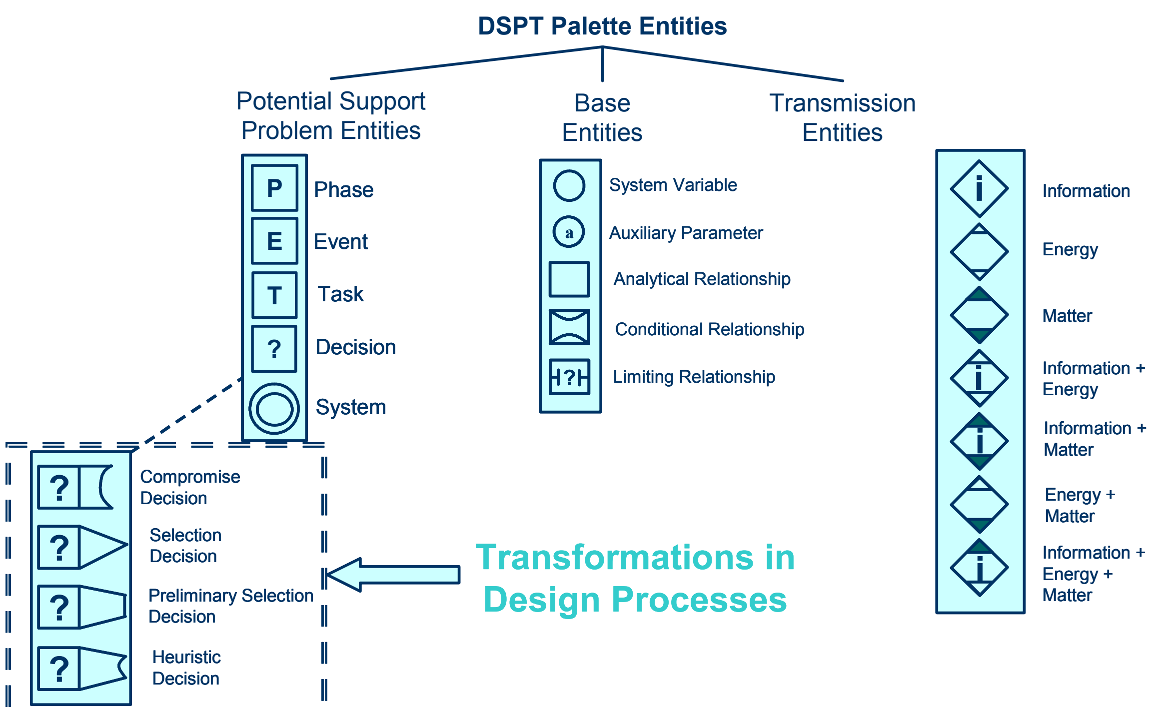


Figure 2-2 - Decision support problem technique palette entities

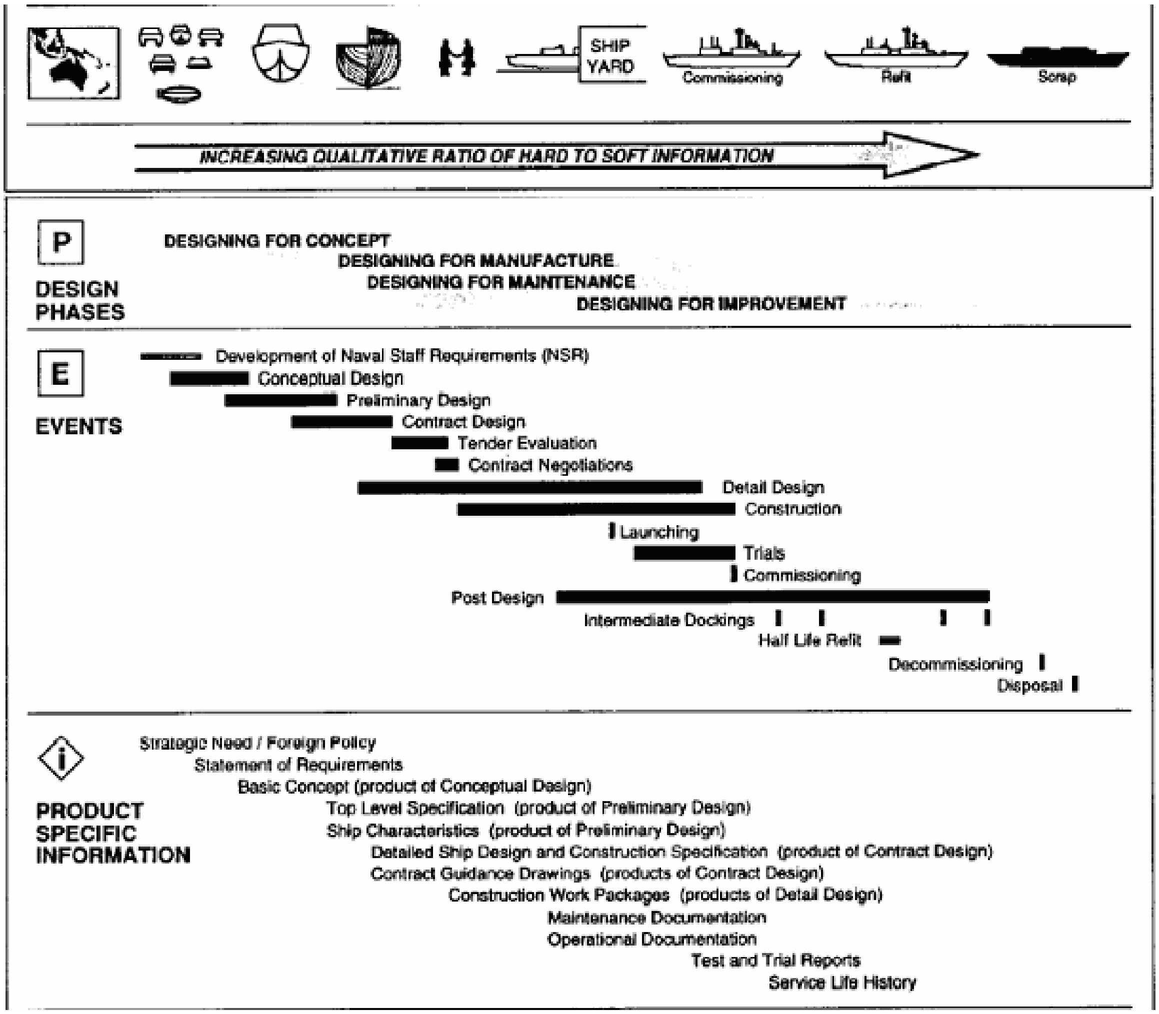


Figure 2-3 Timeline for designing a frigate (Mistree, Smith et al. 1990)

A portion of the timeline of design of a frigate is shown in Figure 2-3. The design phases, events and the information generated is shown in the figure. The qualitative ratio of hard to soft information increases from left to right. To model this design process using the DSP Technique, icons from the DSPT palette are used. The design for concept phase in the process modeled in DSP Technique is shown in Figure 2-4. A detailed description of this design process is provided in (Mistree, Smith et al. 1990).

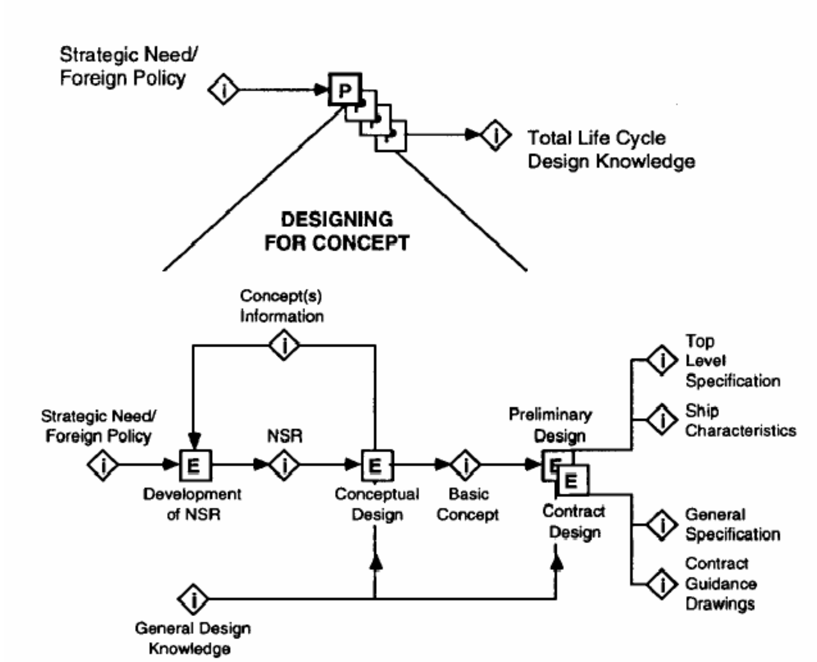


Figure 2-4 A model for conceptual design event (Mistree, Smith et al. 1990)

Using decision support problems for engineering design

The phases (shown in Figure 2-5) involved in using DSPs for designing are:

1. *Planning* – In this phase, the DSPs are identified and summarized in words and figures. This is an unstructured process.

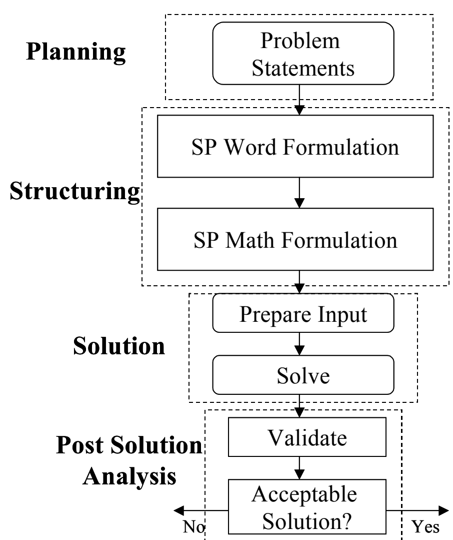


Figure 2-5 Phases in the DSP Technique

2. *Structuring* – In this phase, the qualitative statements from planning phase are transformed into decision support problems that can be stated in mathematical terms

3. *Solution* – In this phase, the designer determines the numerical solution of DSPs.

4. *Post Solution Analysis* – This is a partially structured process of reviewing the design for testing its validity and sensitivity.

Modeling support problems

In order to model the potential support problem entities in a form that can be understood by a computer, the potential support problem entities are associated with corresponding support problems. Within each support problem, the information is organized using keywords and descriptors. The keywords and descriptors act as the medium of communication between a specific designer's view of the world and the domain independent view of the design process. The keywords act on problem descriptors. The descriptors represent domain dependent knowledge and information. The support problems for Compromise and Selection Decision Support Problems are provided in Table 2-2. Each decision support problem has a corresponding mathematical form that contains numerical information about the decision and enables solving the DSP on a computer. The compromise and selection DSP form the core of DSP Technique and are discussed next.

Table 2-2 – Keywords and descriptors for Compromise Decision Support Problem

Keywords	Descriptors
Given	Symbolic and mathematical base entities and Support Problems necessary for evaluating the goals, constraints and bounds and the deviation function
Find	System variables (symbolic and mathematical)
Satisfy	Goals, constraints and bounds, i.e., symbolic and mathematical relationships
Minimize	A Deviation function

Table 2-3 – Keywords and descriptors for Selection Decision Support Problem

Keywords	Descriptors
Given	Alternatives
Identify	Attributes and relative importance of attributes
Rate	Alternatives with respect to attributes
Rank	Order of preference

Decision making in design

According to Kamal (Kamal 1990), all decisions identified in the DSP Technique are categorized as selection, compromise or a combination of these. He classifies selection and compromise as primary decisions and others as derived decisions.

In the DSP Technique, the selection decision is the process of making a choice between a number of possibilities taking into account a number of measures of merits or attributes. The emphasis in selection is on the acceptance of certain alternatives through the rejection of others. The goal of selection in design is to reduce alternatives to a realistic and manageable number based on different measures of merit, called attributes, which represent the functional requirements. The attributes may not all be of equal importance with respect to the decision. Some of the attributes may be quantized using hard information and others may be quantified using soft information.

The compromise decision requires that the 'right' values (or combination) of design variables be determined, such that, the system is feasible with respect to constraints and the performance is maximized. The emphasis on compromise is on modification and change by making appropriate tradeoffs. The goal of compromise in design is that of modification through iteration based on criteria relevant to the feasibility and performance of the system. The emphasis on iteration in compromise which implies that the designer is pursuing a forward progressing process, requiring generation, evaluation and alteration of different designs.

As mentioned previously, all decisions encountered in design can be categorized as either selection among a set of feasible alternatives or the improvement of a given alternative through compromise. Different combinations of these decisions can also occur in design. Decisions can be *interdependent* or *dependent*. Inter dependent decisions involve a bi-directional flow of information and dependent decisions involve a unidirectional flow of information. The flow of information inter dependent decisions is illustrated in Figure 2-6. Since the interdependent (or coupled) decisions involve a bi-directional information flow, both the decision support problems must be solved simultaneously. The dependent decisions arise from cases when in a design timeline, the upstream decisions affect downstream decisions. These dependent decisions must be solved simultaneously.

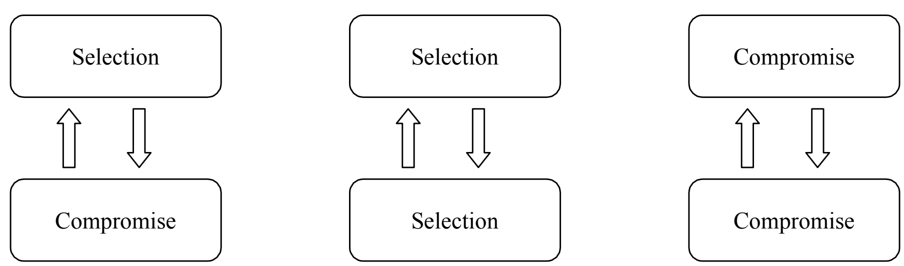


Figure 2-6 Types of interdependent decisions in design

In the DSP Technique, the selection and compromise DSPs are employed to address independent decisions while the coupled DSPs are used to model hierarchies of decisions (Baskaran 1990). The relationship between decisions throughout a design timeline can be modeled in two ways – *hierarchically* and *heterarchically* (Mistree, Smith et al. 1990). In the heterarchical relationship between decisions, the decisions are unordered and it is difficult to define precedence of decisions. In the hierarchical decisions, the information flow is clear and the sequence of decisions is well defined. These decision hierarchies can be implemented using coupled decisions.

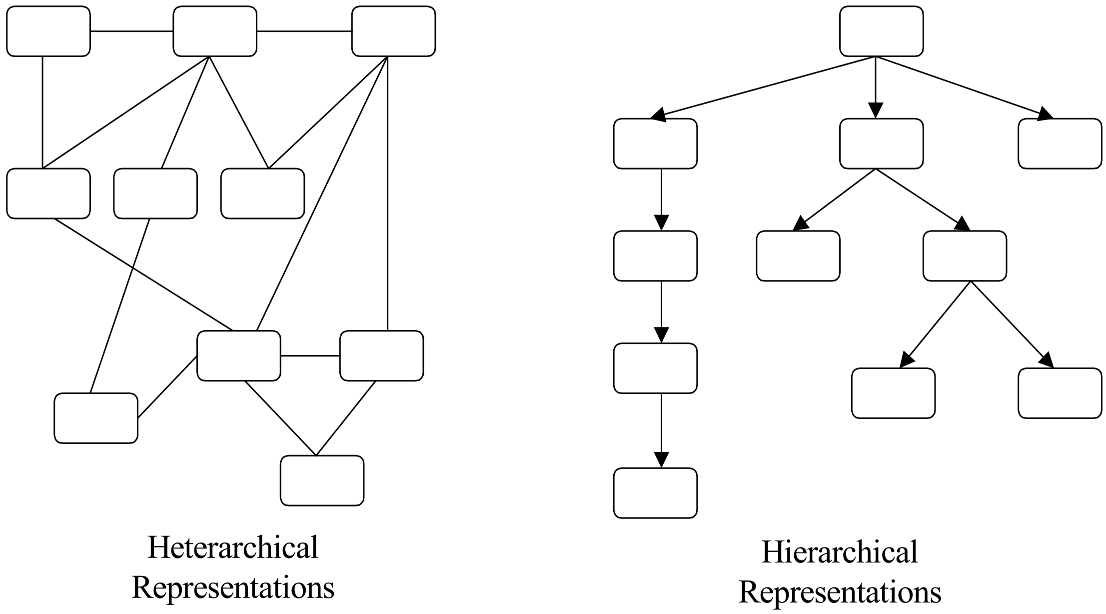


Figure 2-7 Heterarchical and hierarchical representations (Mistree, Smith et al. 1990)

Interdependent or coupled decisions occur in a lot of collaborative and concurrent design scenarios. One of the most common scenarios is the coupling between decisions made by designers and manufacturers. The decisions made by designers affect the decisions made by manufacturers and the decisions made by manufacturers affect decisions made by designers. Coupled decision support problems are applied in various problems like design of composite material structures (Karandikar and Mistree 1993) and components, ship design (Baskaran, Bannerot et al. 1989) etc. Sambu, in his MS thesis (Sambu 2001) has shown a strategy for design for manufacturing based on the coupled compromise DSPs.

Use of DSP Technique in this dissertation

A decision-centric approach is adopted in this dissertation because from a decision-centric perspective, meta-design is a meta-level process of designing systems that includes partitioning the system based on function, partitioning the design process into decisions, and planning the sequence in which these decisions are most appropriately made (Mistree, Smith et al. 1990). Specific advantages of adopting a decision-centric perspective include the ease with which other views of design processes can be generated (e.g., model-centric and tool-centric views). Furthermore, domain independent representation of design processes becomes feasible. Hazelrigg describes decision-based design as omni-disciplinary, "the seed that glues together the heretofore disparate engineering disciplines as well as economics, marketing, business, operations research, probability theory, optimization and others" (Hazelrigg 1998). Herrmann and Schmidt (Herrmann and Schmidt 2002) describe a complete product development organization as a network of decision-makers who use and create information to develop a product. Due to these reasons, a decision centric approach is chosen for design of products and design processes. The utilization of decision-based design in this dissertation is summarized in Table 2-1.

In this dissertation, ***decision-based design is used as a philosophical foundation for the framework for integrated design of products and design processes.*** Hence, the focus of designing design processes and products is mainly on decisions. The first research question in the dissertation – *"How can simulation-based multiscale design processes be designed in association with products?"* is answered from a decision-based perspective. It is assumed that the decisions are the most important components of design processes.

The design of design processes, hence, is equivalent to the configuration of design decisions – related to both products and design processes. We understand that in addition to decisions, there are other activities in a design processes. However, we believe that the decisions represent the most important aspect of design processes and the payoff maximum by considering the decisions. We are not concerned with the lowest level tasks in a design process such as "develop a CAD model", "write a computer program", "document results", etc.

The DSP Technique is an instantiation of the decision-based design, which includes constructs such as compromise DSP, selection DSP for making design decisions in a mathematically rigorous form. This is particularly true in the preliminary design phase, which is the focus of this dissertation, where computer-based simulation models are available. DSP Technique is the only instantiation that is based on designing both products and design processes. ***The meta-design phase of DSP Technique is a foundation for the design method developed in Chapter 3.*** Further, the DSP Technique is based on the idea of making decisions that are *satisficing* as opposed to *optimizing* in nature. The word 'satisfice' was first coined by Simon. It refers to the search for good enough solutions rather than the absolute optimum solutions. The word satisficing should not be viewed as search for inferior solutions because in the words of Simon, "no one in his (/her) right mind will satisfice if he (/she) can equally well optimize; no one will settle for good or better if he (/she) can have the best. But that is not the way the problem usually poses itself in actual design situations." It is the growing complexity of systems that forces designers to search for satisficing solutions. For example, it is not possible to develop *perfect* models of all physical phenomena considering all the interactions with the environment. Because of this approximate knowledge about system's behavior, optimum decision cannot be made. In addition to the lack of perfect knowledge, the environment keeps changing, that renders an "optimal" solution from one scenario "non-optimal" as soon as the environmental conditions change. This is also true in the design of design processes. Due to couplings between different decisions, optimal configuration of design process may not be practical (even if the designer is able to find it).

2.3 Robust Design

Robust design is a design strategy for improving the quality of products and processes by reducing their sensitivity to variations, thereby reducing the effects of variability without removing its sources (Taguchi 1986). Robust design evolved from the research of Genichi Taguchi, who believed that the quality of product should be controlled in the design stage itself, rather than during manufacturing. The fundamental principle behind Taguchi's robust design approach is that any deviation from the target is a loss to the company and represents bad quality of the product. This is in contrast to the generally adopted tolerance based approach where anything that lies between tolerance ranges is acceptable and of equally good quality. Taguchi introduced a Quality Loss Function in which the quality loss, L, is proportional to the square of the deviation of performance, y, from the target value, T.

$$L = k(y - T)^2$$

The objective in Tagichi's robust design is the minimization of this quality loss function over all of the products. This is achieved in the parameter design stage, which occurs before the tolerance design stage. During the parameter design stage, the products are ‘designed for’ robustness by appropriate selection of values for design variables that are least sensitive to variations.

During the parametric design, design parameters are categorized into control factors and noise factors. Control factors are the parameters that designers can control whereas noise factors cannot be controlled easily. Different combinations of values for control variables represent design alternatives. The variations in noise variables result in variation in the performance of design alternative. The best design alternative is the one whose performance is close to the desired performance and the variation in performance is as low as possible. Taguchi used a signal-to-noise ratio to quantify the quality of a design alternative. The design alternative with maximum signal to noise ratio is selected.

Taguchi’s robust design focuses only on performance variation due to noise variables. For the case where control factors are also subject to changes, Chen and coauthors (Chen, Allen et al. 1996) developed a robust design approach that accounts for the objective of minimization of variation in response due to variation in both control factors and noise factors. The authors classified robustness into two types – Type I and Type II as shown in Figure 2-8.

The key difference between the Type I and Type II robustness is:

Type I Robust Design: Identify control factor (design variable) values that satisfy a set of performance requirement targets despite variation in noise factors.

Type II Robust Design: Identify control factor (design variable) values that satisfy a set of performance requirement targets despite variation in control and noise factors.

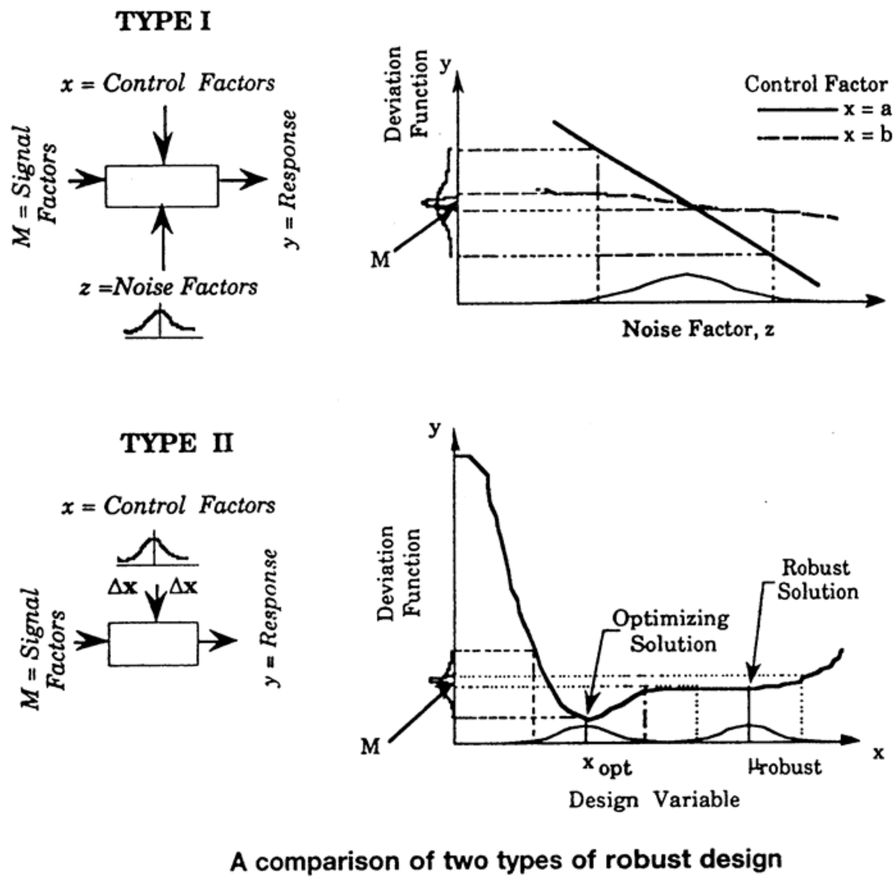


Figure 2-8 – Robust design for variations in noise and control factors (Chen, Allen et al. 1996)

A method combining **Types I** and **II** robust design in the early stages of product development, namely, the **R**obust **C**oncept **E**xploration **M**ethod (RCEM) (Chen 1995; Chen, Allen et al. 1996; Chen, Allen et al. 1996; Chen, Allen et al. 1997; Chen and Lewis 1999) has been developed. RCEM is a domain-independent approach for generating robust, multidisciplinary design solutions. Robust solutions to multi-functional design problems are preference-weighted trade-offs between expected performance and sensitivity of performance due to deviations in design or uncontrollable variables. These solutions may not be absolute optima within the design space. By strategically employing experiment-based metamodels, some of the computational difficulties of performing probability-based robust design are alleviated. In the RCEM method, Chen and coauthors employ the compromise DSP construct as a multi-objective decision model for determining the values of design variables that satisfy a set of constraints and balance a set of conflicting goals, including bringing the mean on target and minimizing variation associated with each performance parameter. RCEM is employed successfully for a simple structural problem and design of a solar powered irrigation system (Chen 1995), a High Speed Civil Transport (Chen, Allen et al. 1996), a General Aviation Aircraft (Simpson, Chen et al. 1996) , product platforms (Simpson, Maier et al. 2001), and other applications (e.g., (Chen, Garimella et al. 2001)).

The idea of robust design is extended to Types III and IV by Choi and coauthors (Choi, Austin et al. 2004). Type III robust design considers sensitivity to uncertainty in the simulation models used to predict the response variables for given design variable values. This type of uncertainty is also called model parameter/structure uncertainty because it exists in the parameters or structure of constraints, meta-models, engineering equations, and associated simulation or analysis models. Choi and coauthors (Choi, Austin et al. 2004) present an approach for Type III robust design by incorporating Error Margin Indices within the basic RCEM. The RCEM-EMI approach incorporates three types of uncertainty – Type I, II, and III. In addition to Type III robust design, Choi and co-authors presented a Type IV robust design approach, which is focused on the uncertainty associated with a design process. Design process uncertainty emanates from the propagation and potential amplification of uncertainty due to combined effect of analysis tasks performed in series or parallel. The sources of design process uncertainty are particularly common and important for multidisciplinary analysis and design, that are characterized by a plethora of shared or coupled variables and analysis performed on multiple length and time scales.

Use of Robust Design in this Dissertation

Robust design is used in this dissertation for making product related decisions. As discussed in the previous section, the decisions are formulated as compromise and selection Decision Support Problems. The design problems discussed in this dissertation are associated with different types of uncertainty that the system should be robust to. These include uncertainty inherent in the environment, uncertainty due to assumptions in the simulation models, uncertainty due to simplification of design processes (caused by ignoring dependencies in the design process), uncertainty propagated from one simulation model to another, etc. Some of these aspects of uncertainty in multiscale design are discussed in Section 1.1.4. The robust formulation of compromise DSP in association with the Robust Concept Exploration method is used in this dissertation to make decisions that are robust to these uncertainties. The design method developed in Chapter 3 consists of robust design techniques discussed in this section as one of its steps. The details of this step are discussed in Section 3.5.4.

2.4 Utility Theory

Utility theory is used to facilitate decision making in product realization based on mathematically complete principles which define 'rational behavior' for the decision makers, and to derive from them the general characteristics of that behavior (Von Neumann and Morgenstern 1947). A decision involves the evaluation of a set of alternatives and selection of the most preferred alternative. 'Utility' represents the decision maker's preference to the outcome, characterized with a set of attributes. In this context, an attribute is equivalent to the response variable that measures the performance of the product. Utility values for attributes generally lie between 0 and 1; a value of 0 denoting an unacceptable outcome and a value of 1 denoting the most preferred outcome. "If an appropriate utility is assigned to each possible outcome and the expected utility of the outcome for each alternative is calculated, then the best course of action is to select the alternative whose outcome has the largest expected utility" (Keeney and Raiffa 1976).

In the terminology of utility theory, a decision is a problem that involves choosing among a set of alternatives X_1, X_2, …, X_n. The consequences of selecting a particular alternative are described in terms of a common set of attributes A_1, A_2, … , A_m. The specific values assumed by these attributes for a particular design alternative X_i are indicated as $A_1(X_i)$, $A_2(X_i)$, … , $A_m(X_i)$. The utilities of the alternatives for a particular attribute A_i are indicated as $u(A_i(X_1))$, $u(A_i(X_2))$, …, $u(A_i(X_m))$, and the utilities of the alternatives considered with all attributes are indicated as $u(X_1), u(X_2), \ldots, u(X_m)$.

If the values of the attributes for different alternatives are known deterministically, then the alternative with maximum utility can be chosen. However, in general design scenarios, the values of the attributes $A_1(\underline{X}_i)$, $A_2(\underline{X}_i)$, … , $A_m(\underline{X}_i)$, for an alternative $\underline{X}_i$ may not be known with certainty, but probabilities can be assigned to the various possible values of each attribute for each alternative. For example, consider a single attribute design scenario, where the designer is concerned only with attribute A_1. If the possible values of an attribute are continuous, the consequences of selecting alternative $\underline{X}_i$ may be characterized by a distribution on the attribute A_1 with an associated probability distribution $f_p(A_1(\underline{X}_i))$ where

$$f_p(A_1(\underline{X}_i)) \geq 0 \quad \text{and} \int f_p(A_1(\underline{X}_i))dA_1 = 1$$

If alternative $\underline{X}_i$ leads to a discrete set of j possible outcomes, a probability p_k can be assigned to each possible outcome where

$$p_k \geq 0 \quad \text{and} \sum_k p_k = 1$$

Given this decision model, a decision maker must select the most preferred alternative when the consequences of each alternative are characterized by probability distributions rather than deterministic values for a set of attributes.

Therefore, given an alternative $\underline{X}_i$, if its attribute A_1 has a certain value, we state that the engineering team's utility for the outcome A_1 is $E[u(A_1(\underline{X}_i))]$. If the attribute has uncertainty, the expected utility for the outcome of A_1 is calculated using its utility function and probability density function $f_p(A_1(\underline{X}_i)$. The expected utility may be calculated as follows:

$$E[u(A_1(\underline{X}_i))] = \int u(A_1(\underline{X}_i)) f_p(A_1(\underline{X}_i)) dA_1$$

If alternative $\underline{X}_i$ leads to a discrete set of j possible outcomes, expected utility is

$$E[u(A_1(\underline{X}_i))] = \sum_{k=1}^{b} p_k u_k (A_1(\underline{X}_i)_k)$$

The basic properties of utilities that are taken as assumptions in the utility theory are:

1) $X_i \succ X_j$ implies that $u(X_i) > u(X_j)$

2) $u(\alpha X_i + (1-\alpha) X_j) = \alpha u(X_i) + (1-\alpha) u(X_j)$

where $\succ$ denotes "is preferred to", X_i and X_j are possible alternatives, and α is the numerical probability that X_i is preferred, (1 - α) is the probability that X_j is preferred.

The maximization of expected utility can be used as a selection criterion only if the utility functions are developed in this manner. If these two properties hold for a utility function, then the utility function is determined to be a linear transformation. The first property implies that the decision maker has a complete set of preferences. That is, given a set of alternatives X_1, X_2, … , X_n, and their attributes $A_1(X_i)$, $A_2(X_i)$, … , $A_m(X_i)$, the engineering team is able to decide a sequence such that $u(X_1) > u(X_2) > u(X_n)$ based on the attributes. The second basic property implies that if activities can be combined with probabilities, then the same must be true with the utilities attached to them. For example, a 50%-50% combination of outcomes X_i and X_j would be the prospect of seeing X_i occur with a probability of 50% and X_j occur with a probability of 50%. Thus the principle indicates that a designer can state whether it prefers the event Xi to the 50%-50% combination of X_j and X_k, $X_i \succ 0.5X_j+0.5X_k$, or vice versa. By answering a set of similar questions, called lotteries, the utility or difference of utilities can be measured.

The question that comes up before using the utility functions for decision making is – *Does a utility function that satisfies the properties discussed above exist?* Von Neumann and Morgenstern postulated three axioms for utility functions. "Provided that these three axioms are satisfied, there exists a utility function with the above properties and with the desirable property of assigning numerical utilities to all possible outcomes such that the best course of action for the individual is the one with the highest expected utility" (Seepersad 2001). These axioms proposed by von Neumann and Morgenstern are shown in Table 2-4.

In the figure, α, β, and γ are probabilities. X's are potential outcomes of a decision. $\succ$ denotes 'is preferred to' and ~ denotes indifference. Axiom 1 is a statement of completeness of preferences and a statement of the transitivity of the preferences. Axiom 2:a states that if X_j is preferable to X_i then even a chance of obtaining X_j is preferable to

X_i. Axiom 2:b is the dual of Axiom 2:a. Axioms 2:c and 2:d are continuity axioms. No matter how desirable an outcome may be, one can make its influence as weak as needed by giving it a sufficient small chance of occurrence. Axiom 3:a states that it is irrelevant in which order the outcomes in a combination are named. Axiom 3:b states that it is irrelevant whether the outcomes are combined in two successive steps with probabilities α, (1-α) and then β, (1-β) or in one operation with probabilities γ, (1-γ), where γ=$\alpha\beta$. These axioms are sufficient to guarantee the existence of a utility function with the desirable property of assigning numerical values to all possible outcomes such that the most preferred course of action is the one with the highest expected utility.

Table 2-4 - von Neumann and Morgenstern Axioms of Utility (Von Neumann and Morgenstern 1947)

The system ***X*** of entities, $X_1, X_2, X_3, \ldots, X_n$ with α and β on the open interval (0,1).

Axiom 1 $X_i \succ X_j$ is a complete ordering of ***X***. This means write $X_j \prec X_i$ when $X_i \succ X_j$.

(1:a) Then for any two X_i, X_j one and only one of the three following relations holds: $X_i \sim X_j$, $X_i \succ X_j$, $X_i \prec X_j$.

(1:b) If $X_i \succ X_j$ and $X_j \succ X_k$ then $X_i \succ X_k$.

Axiom 2

(2:a) $X_i \prec X_j$ implies that $X_i \prec \alpha X_i + (1-\alpha) X_j$.

(2:b) $X_i \succ X_j$ implies that $X_i \succ \alpha X_i + (1-\alpha) X_j$.

(2:c) $X_i \prec X_j \prec X_k$ implies the existence of an α with $\alpha X_i + (1-\alpha) X_k \prec X_j$.

(2:d) $X_i \succ X_j \succ X_k$ implies the existence of an α with $\alpha X_i + (1-\alpha) X_k \succ X_j$.

Axiom 3

(3:a) $\alpha X_i + (1-\alpha) X_k \sim (1-\alpha) X_k + \alpha X_i$.

(3:b) $\alpha(\beta X_i + (1-\beta) X_k + (1-\alpha) X_k \sim \gamma X_i + (1-\gamma) X_k$ where $\gamma = \alpha\beta$.

The process of assessment of utility functions for each attribute consists of three steps: *a)* identification of designer's preferences, *b)* assessment of designer's preferences for levels of attributes, and *c)* fitting a utility function curve with respect to the levels of attributes. The first step involves determining the general characteristics of the designer's preferences. The preferences can be monotonically increasing (larger the better), monotonically decreasing (smaller the better), or non-monotonic as shown in the Figure 2-9. The monotonically increasing utility function can have a convex or concave utility function, depending on the designer's risk taking nature. Convex utility function implies risk aversion while concave utility function implies risk proneness. Same is the case with monotonically decreasing utility function. In a deterministic context, a convex utility function implies that a decision maker has decreasing marginal utility for an attribute at the direction of preference; and in a probabilistic context, a convex utility function implies that a designer is risk averse. The utility functions are determined based on the preference equivalence of two options – the *certainty option*, where the attribute values achieved for different alternatives are known for certain; and an *uncertainty option*, where the designers have 50% probability of achieving the lower bound of attribute values and 50% probability of achieving the upper bound. The expected utilities of these two options are the same. In fact, most of the decisions in engineering design are risk averse.

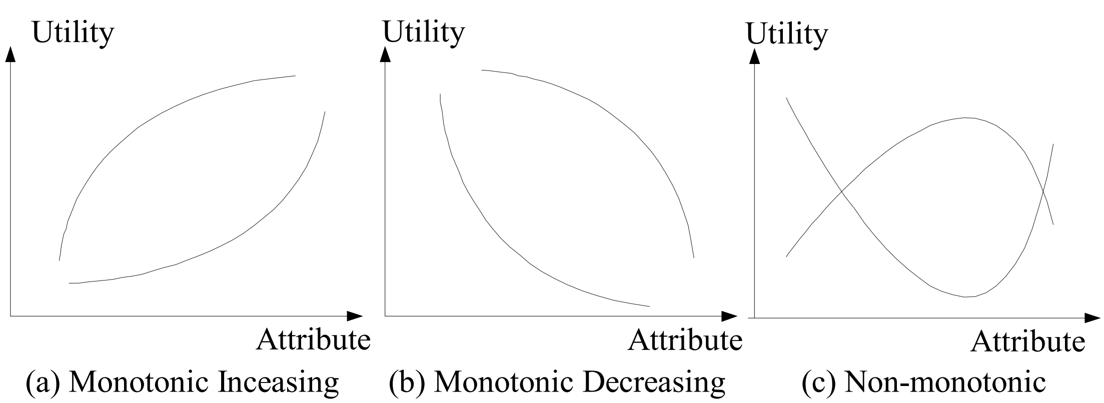


Figure 2-9 - Monotonic and non-monotonic preferences

The discussion so far in this section is focused on single attribute. Design decisions are generally characterized with multiple attributes. The question "how can multi-attribute utility functions be constructed?" is answered by Keeney and Raiffa (Keeney and Raiffa 1976) by developing a method that consists of two stages – a) assessment of utility function for each attribute and b) combination of individual utility functions into a multi-attribute utility function that can be used to evaluate outcomes of alternatives in terms of all the attributes that characterize them. The details of determining multi-attribute utility functions are not discussed in this dissertation. Interested readers should refer Keeney and Raiffa (Keeney and Raiffa 1976). Utility functions are used in this dissertation for modeling designers' preferences because utility theory is a domain independent approach used to facilitate design decision making by evaluating preferences under conditions of risk and uncertainty.

Use of Utility Theory in this Dissertation

In this dissertation, utility functions are used for modeling designers' preferences. These utility functions are used in the compromise and selection DSP constructs for modeling design decisions. The standard compromise DSP is augmented as utility-based compromise DSP is developed by Seepersad (Seepersad 2001). This utility-based compromise DSP is used to model design decisions. The details of steps followed for determining the utility functions are not discussed in detail in this dissertation. Whenever the utility functions are used in this dissertation, it is assumed that a systematic method (that is consistent with the axioms discussed in this section) is followed for determining the utility functions. The focus in the dissertation is on utilizing the utility functions for making decisions and not on eliciting the utility functions, which is a separate research field in itself. Utility functions are also used in this dissertation for making design process related decisions. The process related decisions are made based on the quality of product related decisions. The quality of product decisions is in turn measured as the overall system performance, which is quantified in terms of the overall utility function. Since the second research question in this dissertation (RQ2) is related to developing metrics for quantifying the performance of different design process options, the utility functions form an important component of the hypotheses used to answer RQ-2.

2.5 Information Economics

Information economics is a term for the study of effect of information on decisions. It was first introduced by Howard (Howard 1966). In information economics, the decisions are formulated as follows – the designer wants to select from a specific action a from a set of possible actions $\{a\}$ in the presence of uncertain variable x (Lawrence 1999). The objective is to select an appropriate value of the action $a = a_0$ such that some form of payoff is maximized. The payoff function $\pi(x,a)$ depends on the combination of a, the action that decision maker chooses and x, the ultimate realization of the uncertain variable. The key challenge in decision making is due to the fact that designer has to make decisions before the realization of state x. The designer does not have complete information about the realization of state x. However, some information is available about the probability with which the variable x attains different values. The designer has two options – *a)* he/she can either make a decision-based on the available knowledge about the probability distribution of x, or *b)* he/she can gather additional information that changes the decision maker's probability distribution over the possible values that x can take. This additional information is received from an 'information source'. If the decision maker selects the first option and goes ahead with the first option and makes a decision using the available information about x, then he/she can select the value of action a that maximizes the expected value of the payoff. If the decision maker selects the second option, then he/she must determine whether the cost of gathering additional information is greater than the benefit achieved by gathering additional information. Hence, there is a tradeoff between cost of information and the benefit achieved by it. This tradeoff is quantified by various "value of information" metrics. Most of the "value of information" metrics model the information about benefit achived by additional information only. It does not account for the cost of gathering that information.

Two types of definitions for value of information are particularly important in the information economics literature. These include "ex-post value of information", and "ex-ante value of information" (Lawrence 1999). Ex-post value of information is evaluated by measuring the difference in payoff before and after the realization of states of variable x (but after the decision has been made in both cases). Ex-ante value of information refers to the difference in the expected payoff before and after gathering additional information.

Mathematically, the ex-post and ex-ante value of information are represented as follows:

1. *Ex-post value*: $v(x,y) = \pi(x,a_y) - \pi(x,a_0)$,

   Where, a_0 and a_y represent the actions taken by decision maker in the absence and presence of information y. $\pi(x,a)$ represents the payoff achieved by selecting an action a, when the state realized by the environment is x.

2. *Ex-ante value*: $v(x, y) = E_{x|y}\pi(x, a_y) - E_x\pi(x, a_0)$, where $E_x f(x)$ is the expected value of $f(x)$ and $E_{x|y} f(x)$ is the expected value of $f(x)$ given y. It is important to realize that the key difference between ex-post and ex-ante value is that in ex-post value, the realization of state x is known. However, the realization of state x is not known in ex-ante value and the expected value of payoff is taken over the uncertain range of state x.

Use of Information Economics in this Dissertation

In this dissertation, information economics is used for developing metrics for comparing different design processes based on the overall payoff values that can possibly be achieved by using these processes. Information economics is used for supporting the hypotheses used to answer the second research question RQ2 in this dissertation. This research question is – "How should multiscale design processes be systematically simplified and models refined in a targeted manner to support quick design decision making without compromising their quality". Information economics based metric - Value of Information is used to determine whether a particular level of simplification of design processes is appropriate and whether a particular level of simulation model refinement is right for design decisions at hand. The relation between design process simplification and added information is that when some flow of information is included in a design process, it is equivalent to addition of information for decision making. Similarly, the refinement of a simulation model is equivalent to addition of information for decision making. If the value of this additional information is low, it means that improvement in decision making capability by inclusion of that information is low.

Hence, the simpler design process (or simulation model) is good enough for the decisions to be made.

A review of literature on value of information is carried out to determine whether existing metrics for value of information are appropriate for making design process related decisions. The review of literature is presented in Section 4.2. Based on this review, a set of requirements for value of information for design processes is developed.

2.6 Design Information Modeling

Two key components of design information that have received substantial attention in the design literature include: design processes and product (artifact) modeling. There have been a number of efforts in terms of modeling, representation, archival, exchange, and standardization of information related to engineering products and processes. In this section, we provide an overview of these efforts and identify the need for developing a new information model suitable for addressing the third research question in this dissertation. Section 2.6.1 is dedicated to design processes, whereas in Section 2.6.2, we discuss product modeling related research.

2.6.1 Existing Models for Design Processes

Why do we need to model processes? Modeling processes is an essential step towards building quantifiable models of the product development process. Process models help in documenting the understanding of current ("as is") processes and exploring possible (to be) changes (Lyons, Duffey et al. 1995). They are widely used currently for identifying bottlenecks, activity sequencing and precedence relationships, cost estimation, risk assessment for schedule and cost, archiving the process etc. Currently, the role of design process models is mainly to construct managerially useful decision aids (Smith and

Morrow 1999). Smith and co-authors (Smith and Morrow 1999) categorize the process models into five categories: sequencing and scheduling models, decomposition models, stochastic lead-time models, design review models and parallelism models. Process models are also used for modeling time in the process and minimizing lead-time, and enhancing concurrency between tasks. Process models provide a 'holistic' view of the activities and their relationships. They are widely used for visualizing the flow of information, interactions between activities, resource utilization, etc. In other words, modeling of processes is an essential step towards understanding, analysis and reconfiguration of the processes. Process modeling in general depends on a variety of factors such as: the kind of analysis that the process is subjected to; the knowledge that needs to be archived; the control parameters that control the output of the process and so on. Some of these efforts towards modeling processes and the perspective from which modeling is done are discussed next.

Previous efforts towards modeling processes: Processes can be modeled at various levels of details depending on their intended use. Most of the traditional process modeling methods like PERT, Gantt Charts, IDEF 0, etc. capture information at the activity level. These tools are useful for making organizational decisions on the processes like the time utilization, resource allocation, task precedence, material flow etc. An example of the use of these tools is manufacturing processes modeling to study the time scheduling, material processing, assembly/disassembly and packaging. In a collaborative design scenario, models of processes are needed for understanding and coordinating the collaborative work, defining conflict management (Park and Cutkosky 1999).

Activity-net based models are the earliest and widely used techniques for modeling processes. These activity net-based models are used in two representations: activity-on-node (AoN) and activity-on-arc (AoA). AoN representation is more applicable when the precedence of activities is known where AoA is used when it is important to graphically identify the events in the process. An overview of activity network models is provided in (Elmaghraby 1995). These models are used for analyzing and comparing the complexity of processes, performing risk-based analysis based on the expected time required for different tasks, obtaining critical path etc.

Process specification language (PSL) is an effort at the National Institute of Standards and Technology (Schlenoff, Knutilla et al. 1996) for representation of discrete processes, i.e., processes described as individually distinct events like production scheduling, process planning, workflow, business process re-engineering, project management etc.

These process-modeling methods are generic and can be applied to a variety of scenarios. However, there is a tradeoff between the broadness of applicability of models and the granularity of information that can be represented and the variety of analyses that can be performed on the models. For example, PERT, Gantt Charts, IDEF0, activity-net based models etc. are very general in terms of applicability but can be used to represent information only at an activity and time level. The kind of information being processed is not captured in these models. Hence, it is not sufficient model design processes and to perform the kind of analysis that we want in terms of these models. Since our focus in this research is on design processes, we will look at some of the modeling challenges specific to design processes.

What are the challenges in modeling design processes? Design processes for mechanical systems are complex because of the inherent complexity of the product itself. Interactions and iterations between various activities add to the complexity of product realization processes. Whitney has pointed out in (Whitney 1996) that the complexity of mechanical designs is because of multifunctional nature of the parts to obtain efficient designs. The design processes involve many organizational units and engineering disciplines. The level of human intervention in design process is also a barrier for process modeling. Modeling design processes is also complex because they cannot be completely described beforehand. Downstream activities are highly dependent on the information generated by upstream activities. There is also a high level of uncertainty in these processes. In order to model the design processes, various methods are developed in the literature. These methods can be categorized by the way design is viewed as a process. Some of these views of design processes are design processes as activities, as a decision-based activity, as an evolution of function, as a set of transformations, as a search process, etc. The representation of the design processes is dependent on the view of design process chosen. A summary of approaches to design process modeling is presented in Table 2-5 and discussed next.

Representation of Design Processes using Different Views

There has been a variety of efforts towards modeling design processes from different perspectives. For example, modeling processes from an activity based view (Elmaghraby 1995), functional evolution view (Shimomura, Yoshioka et al. 1998), (Umeda, Takeda et al. 1990), product state evolution view (Hsu, Tai et al. 2000), refinement of product information (Ullman 1992), knowledge manipulation view (Maher 1990), and decision-based view (Mistree, Smith et al. 1990), etc. These efforts are discussed next.

Table 2-5 Comparison of efforts towards design process modeling

Process Modeling Effort	***View of Design***	***Modeling, analysis objective***	***Basic units of a process***	***Comments***
IDEF0 (1993)	Activity based	Organizational decisions	Activities, information	Provides graphical view of the process
DSM (Eppinger, Whitney et al. 1994)	Activity/Task based	Organizational decisions, risk, complexity, probability of rework, iterations, etc.	Tasks	Represents products, processes, identifies interactions, iterations
Shimomura (Shimomura, Yoshioka et al. 1998)	Functional Evolution	Capture design process, designers' intentions, trace design processes	Functional realization, functional operation, functional evaluation	Integrated product and process modeling
Ullman (Ullman 1992)	Evolution of product states	Process representation	Abstraction, refinement, decomposition, patching combination, combination	
Maimon (Maimon and Braha 1996)	Knowledge Manipulation through ASE	Development of a mathematical theory for design	Artifact space, specs, Analysis, synthesis, evaluation	Mathematical representation of transformations
Maher (Maher 1990)	Knowledge manipulation	Development of knowledge based systems	Decomposition, case based reasoning, transformation	Use of artificial intelligence in design
Gorti(Gorti, Gupta et al. 1998)	Knowledge manipulation	Development of engineering knowledge base	Goal, plan, specification, decision and context	Means for representing processes on a computer; integrated product and process representation
DSP Technique (Muster and Mistree 1988)	Decision-based Design	Modeling, analyzing, debugging, finding inconsistencies in a design process	Phases, events, decisions, tasks, information	Formalized templates for decisions, Lacks integration with product information

1. *Activity based view of design:* The design processes when viewed as a set of activities, can be subjected to organizational or scheduling analysis as discussed for the manufacturing processes. Graph based and matrix based methods can be used to represent these processes. The graph-based techniques use activity-net based models. The design structure matrix (DSM) (Eppinger, Whitney et al. 1994) is a popular means for representing both products and processes. Through DSM, we can represent hierarchical structures of both products and processes. The main

advantage of using DSM is the ability to identify interactions and iterations in a design process. Browning and Eppinger in (Browning and Eppinger 2002) use DSM to model processes as a set of activities and process architectures as processes along with their pattern of interaction. DSM is used for a variety of analyses like cost, schedule, risk tradeoffs, probability of rework, level of interactions, complexity, iterations and for process improvement.

2. *Design process viewed as functional evolution:* Shimomura and co-authors (Shimomura, Yoshioka et al. 1998) view design as a process of functional evolution. Design is represented as a process in which a representation of a design object, which includes function, is gradually refined. The representation of design object is based on function-behavior-structure (FBS) model. Each functional evolution involves functional realization (i.e., converting a function into structure), functional evaluation (i.e., confirming functional description with behavior) and functional operation (i.e., adding functional elements and functional relations to functional description). The authors present functional content as a measure of functional satisfaction. One of the advantages of this technique is the ability to model the product (as FBS) and process (as functional evolution) in an integrated fashion. This model can be used to trace the design process and capture designer's intention.

3. *Design as evolution of product states:* A view similar to the functional evolution is the evolution of states of product (Hsu, Tai et al. 2000). The design process is viewed as a problem solving technique by dynamically moving around product state space. A state represents the information about the product at a given point in the

design process. Tomiyama, Yoshikawa and co-workers (Tomiyama and Yoshikawa 1986; Takeda, Veerkamp et al. 1990) view design as a mapping of a point in the function space onto a point in the attribute space. Ullman (Ullman 1992) has also viewed design as a design's initial state and its refinement to the final state. According to Ullman, the essential components to characterize the design processes are: the plan, the processing action, the effect and a failure action. The effects of a design process on the artifact are abstraction, refinement, decomposition, combination, combination and patching.

4. Maimon and Braha (Maimon and Braha 1996) present the use of analysis-synthesis-evaluation (ASE) paradigm for representing design processes in terms of knowledge manipulation. The authors represent the design processes as tuples containing artifact space, specifications and transformation operators: analysis, synthesis and evaluation (ASE). Zeng and Gu (Zeng and Gu 1999) also use model similar to ASE for developing a mathematical model of the design process. The authors develop a basic mathematical representation scheme to define objects involved in the entire design process and investigate design processes with the mathematical representation of design processes. The elements of the design process proposed by the authors include synthesis and evaluation processes, design problem redefinition process, and design decomposition process.

5. *Design as knowledge manipulation:* Another effort focused towards formalizing the representation of design knowledge within the design processes is by Maher (Maher 1990). Maher presented three models for knowledge representation: decomposition, case based reasoning and transformation. The focus of the work was on design synthesis for developing knowledge-based systems. Decomposition involves dividing large complex systems into smaller, less complex subsystems. Case based reasoning involves generation of design solution from a previous design problem. In transformation, the design knowledge is expressed as a set of general transformational rules that can be used in a variety of scenarios.

6. *Decision-based view of design:* Decision-based design is another view of design that has been used for modeling design processes. Mistree and co-authors (Mistree, Bras et al. 1996) view design as a process of converting information into knowledge about the product and decisions are the key markers in the progress of design. Design process can be modeled as a set of decisions. The framework for designing developed with this mindset is the Decision Support Problem (DSP) Technique (Muster and Mistree 1988; Mistree, Smith et al. 1990; Mistree, Smith et al. 1991; Bras 1992; Mistree, Smith et al. 1993; Mistree, Bras et al. 1996). The DSP Technique palette contains entities for modeling design processes. It allows us to arrange and rearrange essential procedures or activities (Mistree, Smith et al. 1990). The entities in the palette are used to build hierarchies and model design processes independent of the design domain (Mistree, Bras et al. 1996). These entities are knowledge and information, tasks, decisions, events and phases. These entities transform information from one state to another as discussed in Section 1. In the DSP Technique, key decision types in engineering are identified: selection, compromise decisions and coupled decisions. These decisions serve as the backbone for modeling design processes. In order to generate information required for executing decisions, supporting tasks are performed.

7. *Object-oriented representation of design processes:* Gorti and co-authors (Gorti, Gupta et al. 1998) have developed object-oriented models for design processes and products. The key elements of a design process modeled by the authors in the design process are goal, plan, specification, decision and context. The design artifact includes function, behavior, structure, and causal knowledge relating objects to physical phenomena. Primary objective of the authors is to develop comprehensive engineering knowledge bases and the effort was not focused towards analysis of the process.

Summary of design process representation using different views: These efforts towards modeling design processes are summarized in Table 2-5. It is important to note here that there is no single design process model that encompasses all the aspects of design. Some of the methods are focused on capturing processes to make organizational decisions, some towards understanding and capturing designers' intentions, while others are focused towards artificial intelligence. Modeling is essentially representing a view of the real world. Hence, it is important to understand which view of the world is important in a given scenario. The view that we are interested in is: "the manner in which product information evolves". Most views that researchers have taken towards modeling design processes are managerial in nature. Our focus is on providing design support at a designer level and not so much at a managerial level.

Mathematical models for design processes

In addition to the representation of design processes from different perspectives, a few efforts have also been made towards representing design processes as mathematical equations. These efforts are important from the perspective of designing design processes because they formalize the design processes in a form suitable for mathematical analysis. In order to design the design processes, this is a fundamental prerequisite - to model design processes in a manner that they can be analyzed with respect to the impact of individual transformations on the product information, thereby providing a mathematical understanding of design evolution. Such an analysis of design processes is possible only if we can represent these processes in a mathematical form. Due to their importance in this context, they deserve a review in this section.

Braha and Reich argue that the two main reasons of casting design in mathematical terms are – *a)* mathematical models of design improve the understanding of limits of formalizing design and the limit of automating it, and *b)* studying the mathematical model of design could produce practical guidelines or ideas for implementing design support procedures or systems (Braha and Reich 2003). In this dissertation, we approach mathematical modeling of design processes from a standpoint of developing understanding and building strategies for designing design processes.

Key efforts towards mathematical representation of design processes in terms of design equations include design equations developed separately by Suh (Suh 1990) and Bras (Bras). These efforts to mathematical modeling of design processes as equations are discussed next.

1. Suh (Suh 1990) presented a design equation $\{\mathbf{FR}\} = [\mathbf{A}]\{\mathbf{DP}\}$, which represents a mapping between functional domain and physical domain. Functional domain refers to what we want and the physical domain refers to the means for satisfying what we want. Functional Requirements (**FR**) in the design equation refers to the minimum set of independent requirements that completely characterize the functional needs of

the product design in the functional domain. Design Parameters (**DP**) are the key variables that characterize the physical entity created by the design process to fulfill the **FR**s. The matrix [A] in the design equation is called the *design matrix*. A similar vector equation can be written for manufacturing processes that maps physical domain to process domain.

The structure of the design equation provides insight into the quality of design – i.e., whether the design is coupled, decoupled, or uncoupled. In an uncoupled design, the design matrix [A] is diagonal and each functional requirement is satisfied independently by design parameters. When the matrix is triangular, the design is decoupled, whereas any other form of matrix refers to a coupled design. The linear design equation is valid for only a given level of abstraction. Design can be represented at various levels of abstraction using the idea of zigzagging. Zigzagging refers to successive application of *a)* mapping of functional requirements to design parameters and *b)* decomposition of functional requirements to concrete subsets that can be mapped to design parameters at a lower level of abstraction (greater detail).

Suh's design equation is an embodiment of the two design axioms – independence axiom and information axiom, and forms a basis for axiomatic design. Hence, it serves as a guideline for what a good design is. Using the design equation, it is possible explain and quantify concepts such as coupling in design. Further, Suh's design matrix shows what the characteristics of design should be in order to facilitate concurrent design (the overall product of design matrix should be decoupled – diagonal or triangular).

The design equation proposed by Nam Suh's serves as a guide to the designer in differentiating good and bad designs. However, it is unable to capture -

a) Complex relationships between entities of product information. This is because there is no information model associated with the design equation to represent FRs and DPs. Hence, relationships between parameters, functional requirements etc, are not captured.

b) Activities (transformations) in design process other than mapping. The *only transformation* of information captured by the design equation is mapping from one domain to another. None of the other activities carried out in designs such as decomposition, abstraction, evaluation, etc are modeled as transformations in the design equation.

c) The mathematical design equation proposed by Suh is not directly amenable to computational implementation of a design process.

Suh's design equation is primarily used for making decisions about the product (which design is better), not about the process. Suh's design equation serves as a guideline for what a good design is but it does not provide any guidelines for designing design processes (especially since is only tied to one level of abstraction). The aspects related to meta-design such as - when to decompose, how to decompose, which design process path to select, etc. are not answered. Further, the design equation cannot be used to understand *how the product information evolves along the design process*. Reusability of design process related information is not addressed in Suh's design equation.

2. Bras and Mistree (Bras) developed a generalization of Suh's design equation. In their design equation, a single transformation in design process is represented as an algebraic design equation, namely **K=T(I)**, where **I** is a vector with *n* components representing information and **K** is a vector with *m* components representing the knowledge. **T** is a vector function transforming the vector **I** into **K**. Suh's design equation is a special case of the design equation developed by Bras and Mistree because Suh's design equation captures only a linear transformation from one space to another, whereas, the design equation by Bras and Mistree captures non-linear transformations also.

   The meta-design equation is represented as $\Delta K = [\mathrm{T}]\Delta I$, where $\Delta I, \Delta K$ represent difference in information and knowledge respectively. The DSP Technique allows formulation of the transformation function **T**() in terms of Decision Support Problems solvable on a computer. The focus while developing the design equation is on meta-design (designing design processes). The coupling between meta-design and design is shown as the following equation –

$$\mathrm{T}_{ij} = \frac{\partial(\mathrm{T}_i(I))}{\partial(I_j)}$$

   This equation represents the need for integrated design of products and design processes. The meta-design and design equation represent only single transformations of information into knowledge. In order to capture the *process* consisting of multiple transformations of information into knowledge, the notion of a *sequence* is introduced. The *design process equation* is represented as –

$$\{\mathbf{K_k}\} = \{\mathbf{T_k}(\{\mathbf{I_k}\})\}$$

   where $\{\mathbf{I_k}\}$, and $\{\mathbf{K_k}\}$ are sequences of k vectors.

Note that since the process is represented as a *sequence* (read as set) of transformations, the connectivity of information flowing through the design process is not preserved. Again, the manner in which product evolves is not clear. Also, the relationships between elements of product information are not maintained.

One of the advantages of design and meta-design equations is that they capture design history. As pointed out by Bras, the fundamental question to be answered in design of design processes is – "*What is the best formulation of the design equation?*" Bras used compromise DSP to design the design equation. "The compromise DSP for designing the design equation represents the highest level of decision making within the DSP Technique". As in the case of Suh's design equation, the compromise DSP can be applied recursively. The notion of hierarchy is inbuilt in design. The contents of the formulation of DSPs may change but not their structure. The transformation is a relationship that can occur in different forms, such as, computer program, knowledge base, rule network, etc. Hence, it is more suitable for computer implementation. Note – there is a difference between matrix [T] and function T(). DSP Technique uses function, whereas Suh uses matrix. At the meta-design level, however, matrix notation is used because we are considering a linear approximation. The design equation is left only at this level of detail.

3. In Formal Design Theory (FDT) (Braha and Maimon 1998), the design processes are represented as a finite automaton $DP \equiv \langle L, Q, P, T_A, T_S, S_0, F \rangle$, where L denotes the design description, Q is a set of finite process states, P is a set of production rules, T_A is the analysis transformation, T_S is the synthesis transformation, S_0 is

the initial process state, and F is a set of terminal process state. The design process involves a series of transformations that transform one process state to another. A process state is described by $S \equiv \langle M, \theta \rangle$, where, M denotes the artifact description, and θ denotes specifications. The series of transformations ends with a terminal process state. A transformation, which is described by two adjacent process states, is activated by a set of knowledge tokens in the form of rules included in the designer's knowledge body. Later, in (Braha and Reich 2003), Braha and Reich represented the design processes as topological transformations in the design space. One of the important advantages of Braha's model (Braha and Reich 2003) is that it explains abstraction and refinement as mathematical (topological to be specific) operations. None of the other models are capable of doing that.

4. Zeng and Gu (Zeng and Gu 1999) presented a mathematical model of the design process. The authors develop a basic mathematical representation scheme to define objects involving the entire design process and investigate design processes via their mathematical representation. The design governing equation proposed by the authors is -

$$S = K^s(\lambda * (K^p(S)))$$

where, S is the product description, K^s is the synthesis knowledge, λ is the design specification predicate which denotes whether the structural and performance constraints are satisfied or not, and K^p denotes the property (behavioral) knowledge that can be derived from product simulation. This is a recursive design equation because the product description (S) is defined in terms of itself. The design process is essentially the process used to solve this recursive equation. The mathematical model for design process is illustrated in a graphical form in Figure 2-10. The authors propose a six step prescriptive method to solve this design equation by assuming that there are some primitive products that are known to the designers and the knowledge about their behavior exists. The steps in solving the recursive design equation include the following: synthesis and evaluation, design problem redefinition, and design decomposition. Each of these activities is defined in terms of set theory. The complete design process is defined in terms of a set-theory based mathematical language. The authors also proposed an accompanying mathematical model for the representing the product information.

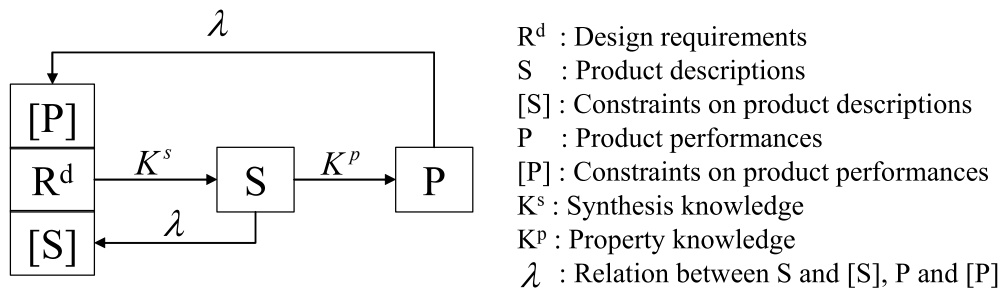


Figure 2-10 - Design process model proposed by (Zeng and Gu 1999)

Comparison of existing process modeling efforts

From the review of literature on mathematical modeling of design processes, we observe that the two categories of design process modeling efforts (mathematically rigorous models and computer-based models) are being pursued rather independently. We envision that by integration of these two streams of modeling efforts, more effective design support tools can be developed. A comparison of four modeling efforts which are most applicable to the focus of this chapter is presented in Table 2-6. Activity based models are selected to represent a variety of modeling efforts such as DSM, PERT and

Gantt charts, etc. The three efforts - Zeng and Gu (Zeng and Gu 1999), Suh (Suh 1990), and DSP Technique (Bras) are selected because of their emphasis on mathematical modeling of design processes.

Table 2-6 – A comparison of the key modeling efforts in design

Requirements for Design Information Modeling Approach		Activity Based Models	Zeng and Gu (Zeng and Gu 1999)	Suh (Suh 1990)	DSP Technique (Bras)
A) Support for designing design processes					
1.	Existence of mathematical models for design processes and products	✗	✓	(not for products)	(not for products)
2.	Ability to model linkages between mathematical model and computational model to support execution	✗	✗	✗	✓
3.	Support for design decision making (the information model should capture designers' preferences, goals, etc.)	✗	✓	✗	✓
4.	Ability to define design problems	✗	✗	✗	✓
5.	Ability to identify better designs and suitable courses of actions	✓	✓	✓	✓
B) Modeling process information					
6.	Capability to define processes at all these levels of abstraction	✓	✓	✓	✓
7.	Support for Composability of sub-processes into overall processes	(not at a computational level)	(not at a computational level)	(not at a computational level)	(not at a computational level)
8.	Separation of problem formulation from process information and tool specific execution details	✗	✗	✗	(allows separation of problem and process, but not tool specific information)
C) Modeling product information					
9.	Capability to understand the evolution of product information along the design process	✗	✓	✗	✗
10.	Ability to generate meta-information about the design space	✗	✓		
11.	Ability to represent uncertain information	✗	✗	✗	✗
D) Reuse of information					
12.	Support reusability of processes at computational level	✗	✗	✗	✓
13.	Modular use of processes for different products	✗	✗	✗	✗
14.	Modular use of processes for different design problems	✗	✗	✗	(not at computational level)

As can be seen in Table 2-6, DSP Technique is the approach that is most suitable for addressing requirements presented in the previous section. This is due to a greater emphasis on a) meta-design, b) mathematical modeling of processes using design equation, and c) integration of mathematical models with computational models. Other efforts focus mainly on mathematical modeling, but not on computational execution of processes. Hence, DSP Technique is used as a basis for development of the proposed 3-P approach. The details of the DSP Technique and the associated constructs useful for meta-design are discussed in Section 7.2.1.

2.6.2 Existing Product Modeling Methods

In addition to the design process modeling efforts, a number of efforts are also focused on modeling product information. The product models are relevant in the context of meta-design because the activities in a design process transform the product information from one state to another. Hence, the product information becomes an operand in the design equation. This section is devoted to the discussion of significant efforts towards developing product models. From this literature review, we identify the requirements for the product models to support meta-design. Similar to the process models, the models for product information can also be categorized into computer-based representations and mathematical models. Examples of mathematical product models include the models discussed in references (Braha and Maimon 1998) and (Zeng and Gu 1999; Zeng and Gu 1999). There is a slew of computer-based representation of product information ranging from CAD formats to object oriented models. Some examples of computer-based representation of product models are presented in references (Gorti, Gupta et al. 1998), (Eastman, Bond et al. 1991), (Nell 2003; Peak, Lubell et al. 2004), and (Fenves 2001). Some of these modeling efforts are discussed next to provide a flavor of the available literature.

Braha and Maimon (Braha and Maimon 1998), in their Formal Design Theory, develop an artifact representation consisting of a multiplicity of modules and relationships among them. A design at any particular level of abstraction is a description of organized collection of constraints that are to appear in the physically implemented design. The artifact space is described as an algebraic structure of modules that are characterized as either basic or complex modules. An artifact space is a tuple $<M_0, C^0, M^*>$, where M_0 represent a set of basic modules, M* represent a set of complex modules that are developed by combinations of basic modules. The relationships between modules are represented as constraints C^0. The complex modules can be used to represent hierarchy in product models. Braha and Maimon's model is similar to the entity relationship models. The entities and modules are physical objects.

The 'Integrated Product and Process model' developed by Zeng and Gu (Zeng and Gu 1999; Zeng and Gu 1999) consists of a mathematical description of the product. The product description consists of a set of basic elements and combination rules. These basic elements are the primitive components whose performance can be obtained independent of the other components. Examples of such basic elements in a mechanical design scenario include gears, shafts, bearings, etc. The assumption used for modeling product information is that any product can be broken down into sub-assembly components and their relationships. The basic elements are described in terms of a set of structural properties and associated values. An example description of a spring is {<d0, 25.0>, <d, 2.0>, <d1, 21.0>, <L0, 234>, <Na, 11>, <Ni, 11>, <p, 20.0>}, where d0 is the outside diameter, d is the wire diameter, d1 is the inside diameter, L0 is the free length, Na is the number of active coils, Ni is the number of inactive coils, p is the pitch, and m is the material. The connectors are represented in a manner similar to the product representation. Using this description for basic elements, complex products are defined as a set of components related to each other through component connectors. This introduces a hierarchy in the product description.

Gorti and co-authors (Gorti, Gupta et al. 1998) developed an object-oriented representation for product and design processes. The objective in the representation is to develop a representation for engineering knowledge-bases in a layered fashion. The knowledge base is developed such that it supports stepwise refinement. The key aspects of the artifact represented in the product model include function, form, and behavior. The information model supports representation of constraints in the artifact information. The representation allows for comparison of similar objects in a meaningful way.

Fenves (Fenves 2001) developed a basic information model for representing product information that is based on the function-structure-behavior view of design. The model is very general and serves as a foundation for development of domain specific information models. Eastman and co-authors (Eastman, Bond et al. 1991) developed a formal approach to product model information where the objective is to develop engineering databases schemas. The formal product modeling approach supports multiple levels of abstraction of information, provides support for form and physical properties, representation of semantics, provides capability to represent dependencies (database integrity) and constraints. The data-models are defined bottom-up from sets and predicate logic. The formal language used in the information modeling approach is based on first order predicate calculus. Other efforts such as *Standard for Exchange of Product Information* (STEP) are focused on improving data exchange mechanisms for engineering and design (Nell 2003; Peak, Lubell et al. 2004).

Based on the models described in this section, it is apparent that there are at least two common strategies adopted in each of these models. The first strategy is the adoption of function-behavior-structure model and the second strategy is the use of entity-relationship models for information representation.

Use of Information Modeling in this Dissertation

The role of information modeling in this design is highlighted in Table 2-1. ***Information modeling constructs are used in this dissertation to provide computational support for meta-design***. In this dissertation, we use and develop the existing information modeling constructs for enabling the current simulation-based design frameworks to support meta-design. The constructs for product and process modeling are used for answering the third research question in the dissertation that is related to supporting meta-design in a computational environment.

2.7 Role of Chapter 2 in the Dissertation

The objective in this chapter is to introduce the fundamental constructs based on which the framework for integrated design of products and design processes is developed. Five fundamental constructs discussed in this dissertation include *a)* decision-based design and DSP Technique, b) robust design, c) utility theory, and d) information economics, and e) information modeling in design. These fundamental concepts are used throughout the dissertation, with references to appropriate sections in this chapter. The utilization of these constructs in the development of different components of the framework is highlighted in Figure 2-11.

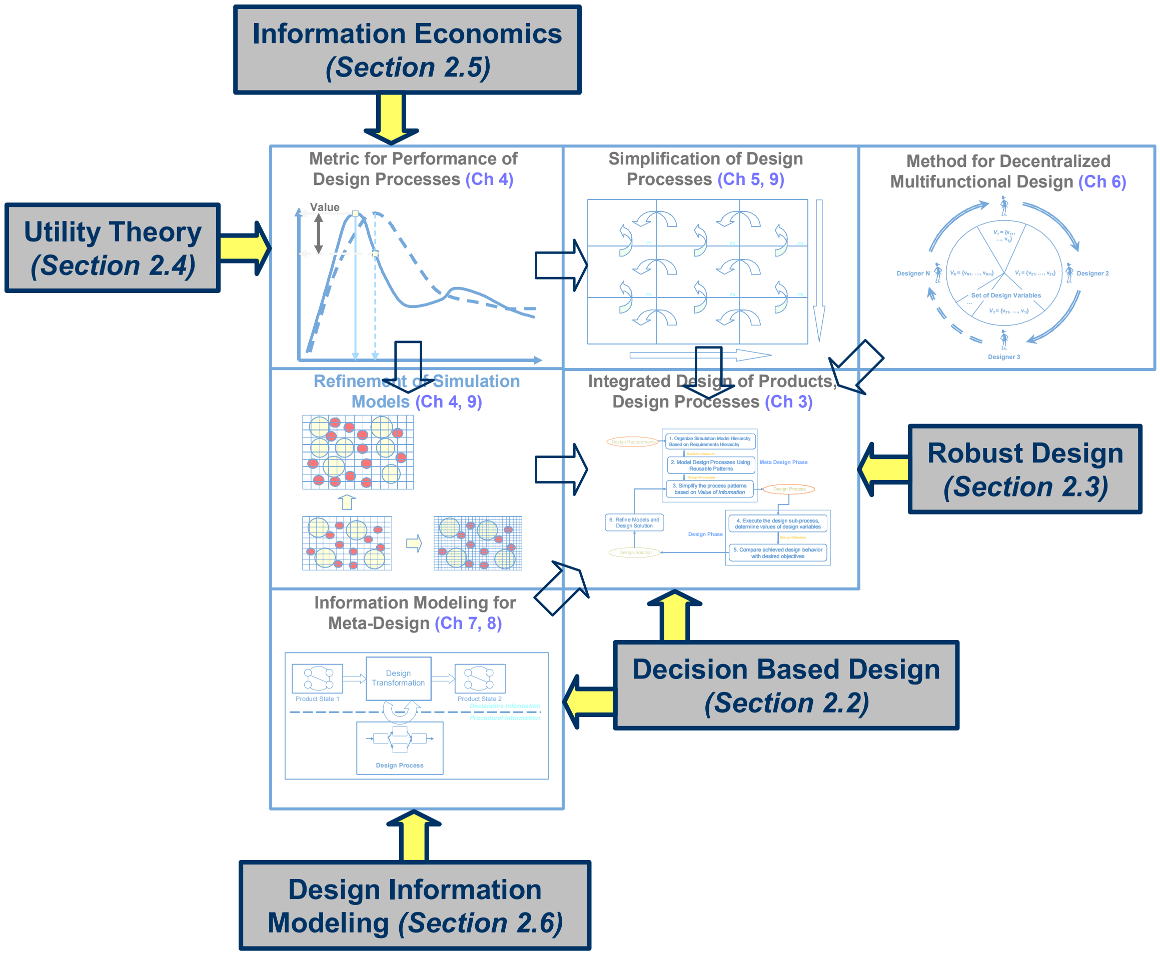


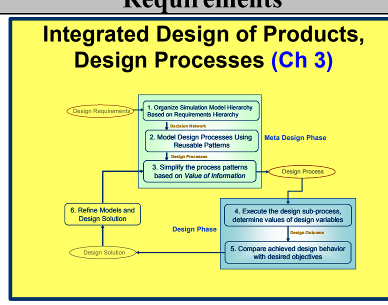
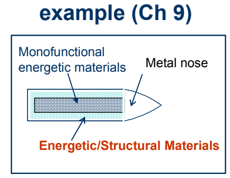
Figure 2-11 – Utilization of constructs presented in this chapter for a framework for integrated design of products and design processes

Chapter 3 A Method for Integrated Design of Multi-scale Products and Associated Design Processes

3.1 Context – Answering Research Question 1 - Designing Design Processes

In this chapter, we present the first component of the framework: *a design method for integrated design of products and design processes* (see Table 3-1). The method is called a Robust Multiscale Design Exploration Method (RMS-DEM). The discussion in this chapter is focused on answering the first research question in this dissertation – "How can simulation-based multiscale design processes be designed in association with products?" This research question is supported by two hypotheses – "systematic, stepwise refinement of design processes and associated products increases the efficiency and effectiveness of design decision making (H1.1)", and "design processes can be designed as hierarchical systems composed of repeating building blocks defined in terms of interaction patterns (H1.2)". These two hypotheses are embodied in the design method presented in this chapter. The method is validated in Chapter 9 using an integrated materials and product design. The requirements, component of the framework and the validation example related to this chapter are shown in Table 3-1.

Table 3-1 – Requirement of the framework addressed in Chapter 3

<table>
<tr><th>Framework Requirements</th><th>Components of the Framework Developed to Address the Requirements</th><th>Validation Examples</th></tr>
<tr><td>1) A method for integrated design of products and design processes</td><td>
</td><td>Materials-Product design example (Ch 9)


Purpose: To validate the method for integrated design of products and design processes</td></tr>
</table>

The design method consists of two phases – *meta-design* and *design*. In the *meta-design* phase, the design process is designed and in the *design* phase, the process is executed to design the product. The two phases are carried out in a cyclic fashion with successive refinement of the design processes along with the refinement of the associated product (see H1.1). The second hypothesis (H1.2) is embodied via identification of reusable building blocks of design processes. These building blocks are defined in terms of interaction patterns between simulation models and between decisions. The research questions and hypotheses addressed in this chapter are highlighted in Figure 3-1. A Multifunctional Energetic Structural Material design example is used for validating the method presented in this chapter. The validation example is discussed in Chapter 9.

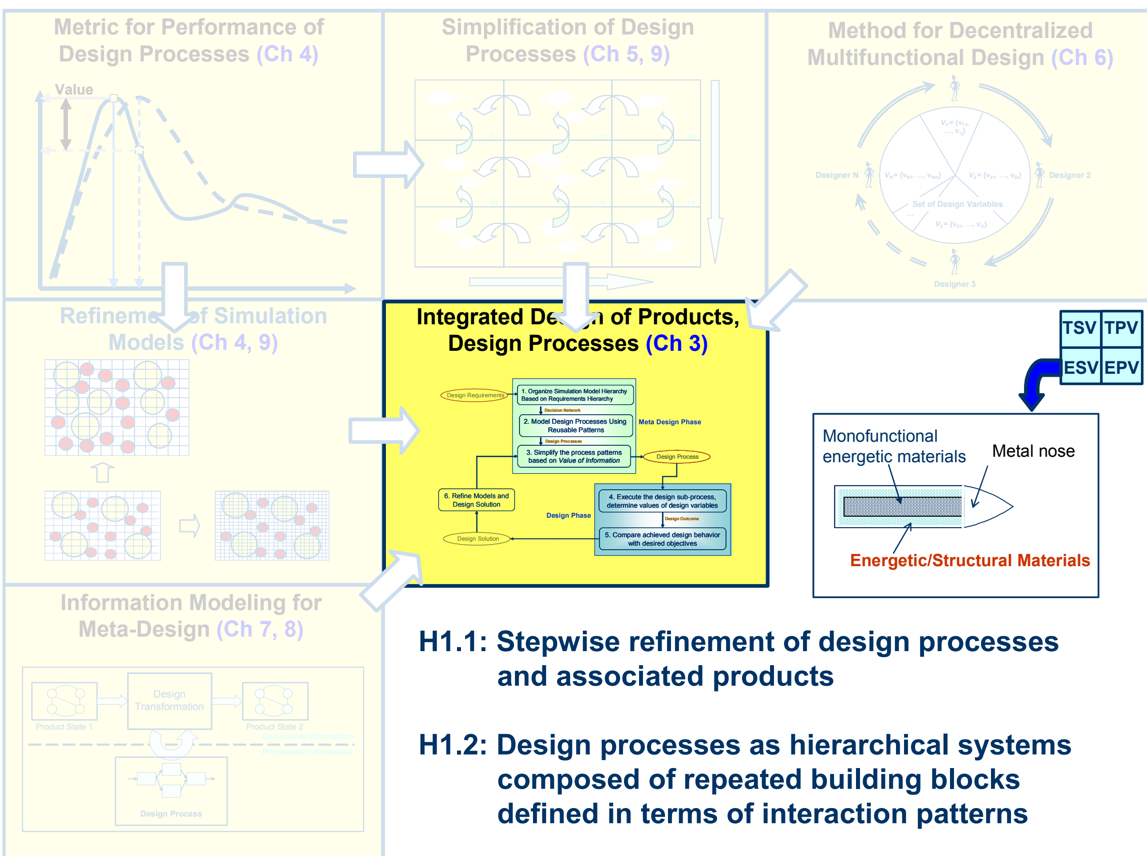


Figure 3-1 –Hypotheses addressed in Chapter 3

A pictorial overview of this chapter is presented in Figure 3-2. In Section 3.2, a brief overview of the strategy adopted in the design method is presented. The strategy consists of four foundational elements that form the basis for integrated design of products and design processes. For designing the design processes, it is important to be able to model them in a manner that supports their analysis and reconfiguration. Hence, before discussing the method for integrated design of products and design processes, it is important to discuss models used for representing design processes. The aspects of modeling design processes are discussed in Section 3.3. Existing literature on modeling and design of design processes is discussed along with the identification of needs that should be addressed in the design method. After discussing the gaps and requirements, a simple motivational example of design of structures is presented in Section 3.4. The structure design problem is used as a running example to demonstrate different aspects of the method, where it is discussed in Section 3.5. The method consists of six steps, each of which is discussed in Sections 3.5.1 through 3.5.5.

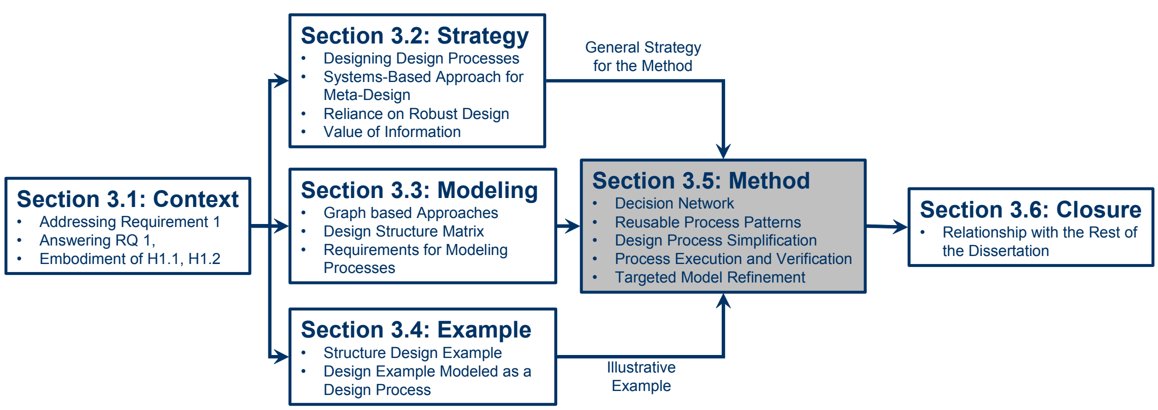


Figure 3-2 – Pictorial overview of Chapter 3

3.2 Strategy for Integrated Design of Products and Design Processes

In this section, we present our strategy for integrated design of products and design processes, which is based on the four foundational elements: *1)* designing design processes in association with products, *2)* applying systems-based approach for meta-design, *3)* reliance on robust solutions, and *4)* systematic refinement of design. These elements of the strategy are discussed next.

1. *Designing design processes:* The design processes for complex systems are often structured in an ad-hoc fashion based on the previous design experience. However, due to the complexity of multiscale systems, it is imperative that the design processes themselves be designed systematically based on the design problem at hand. This is because inefficient design processes can lead to longer design time, thereby leading to higher costs. This was also pointed out by Herbert Simon as "design process strategies can affect not only the efficiency with which resources for designing are used, but also the nature of final design as well" (Simon 1996). The design of design processes thus constitutes a necessary step for efficient utilization of information available from simulation models at multiple scales. Hence, the proposed design method consists of two phases – a) ***meta-design*** and b) ***design***. During the meta-design phase, the design processes are designed and during the design phase, these design processes are executed (Bras and Mistree 1991). This two-phase design method is shown in Figure 3-8. The method consists of six steps. These steps are discussed in Sections 3.5.1 through 3.5.5.
2. *Systems-based approach for* ***meta-design****:* Our approach for designing design processes involves thinking about processes as systems that can be partitioned into

sub-systems with clearly defined interfaces. From a hierarchical systems standpoint, design processes can be progressively broken down into sub-processes that can be further represented in terms of basic design process building blocks, namely the information transformations. Specifically, we identify standardized *design process patterns* with well defined inputs and outputs that facilitate hierarchical modeling of design processes. These process patterns are captured as reusable templates used to model any multiscale design process. The design processes, modeled in such a manner, provide the ability to reconfigure them. Specific design process patterns identified in the case of multi-functional design processes are discussed in detail in Section 3.5.2.

3. *Reliance on robust* ***design****:* Simulation models at multiple scales are characterized by uncertainty that should be taken into account during design decision making. Hence, the design methodology proposed for multiscale design is based on robust ranges of solutions instead of optimum point solutions. Consideration of robust ranges of solution is also important when the requirements change along the design process.
4. *Value of information metric for systematic refinement and simplification of* ***design****:* In order to increase the efficiency of design processes, it is important that analysis model development and design exploration be carried out in a parallel fashion, instead of developing all simulation models to completion and then executing starting the design process in a sequential fashion. The objective here is to maximize design process efficiency by narrowing down the design space in the preliminary design phase by using approximate models and then performing finer design exploration using exact models. This is very important when the simulation models evolve with time.

Using these four guiding principles, we develop a method for the integrated design of products and design processes. Meta-design is one of the most important aspects of the method. Section 3.3 is dedicated to understanding the fundamental aspects of modeling and designing design processes. The remaining part of this chapter is focused on the method itself and a structural design example to demonstrate the utilization of the method.

3.3 Modeling and Designing Design Processes

In Section 3.3, we discuss *a)* current methods used to model design processes that support making meta-level decisions (Section 3.3.1), *b)* existing literature on designing design processes (Section 3.3.2), which is primarily available in the context of concurrent engineering, and *c)* our view of design processes – networks of information transformations (Section 3.3.3). These design process modeling and meta-design approaches are augmented and leveraged in the method developed in this Chapter in Sections 3.5.1 and 3.5.2.

3.3.1 Modeling Design Processes – Network and Matrix-Based Approaches

One of the simplest approaches used to represent design processes is the *directed graph* approach. These graphs are constructed with activities represented as nodes that are interconnected by arcs that represent information flows between activities. The arcs are directional relationships showing the source and sink of information flows (see Figure 3-3(a)). Directed graphs serve as a good visualization tool for understanding the interdependencies between design tasks. They are used for simple design processes where there are only a few activities. They are not useful for representing complex processes because the directed graphs get cluttered easily with increase in number of nodes.

Another limitation of the directed graphs is their inability to express temporal precedence.

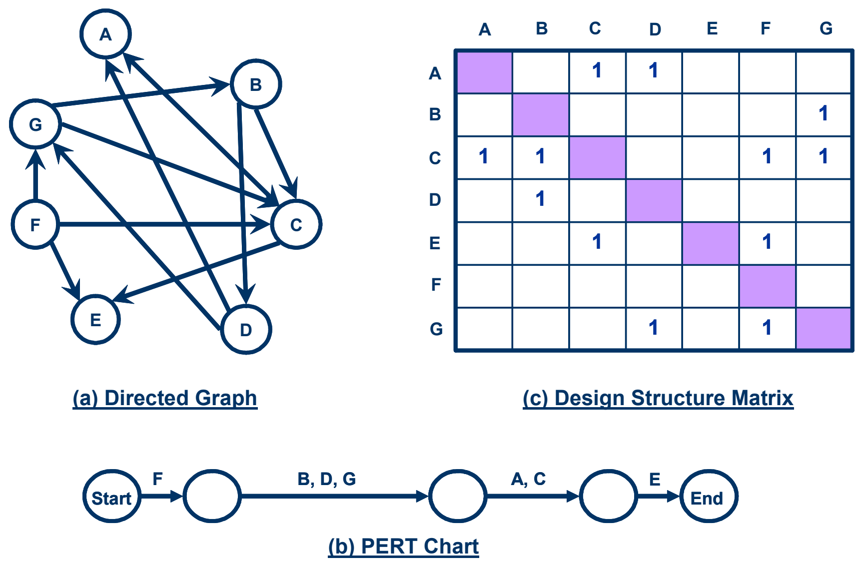


Figure 3-3 – Directed graph, PERT chart and Design Structure Matrix for modeling design processes (Gebala and Eppinger 1991)

A modification of the directed graph approach is a PERT (Program Evaluation and Review Technique) chart, where the nodes of a directed graph are arranged along a timeline (see Figure 3-3(b)). In PERT charts, the tasks are placed along the arcs and nodes represent project milestones. The arc-lengths are proportional to the time required for execution of the design tasks. These PERT charts are generally used in scenarios where there are multiple ways of performing a design task, and the objective is to determine the path that requires minimum amount of time (i.e., critical path). PERT charts are popular in manufacturing processes. The main limitation of PERT charts is their inability to represent design iterations that are represented as loops in the directed graphs.

In addition to the graph based approaches for representing design processes, matrix-based approaches are widely employed for modeling design processes. The matrix representation of processes is termed as the Design Structure Matrix (DSM) and was introduced by Steward (Steward 1981). A design activity composed on n tasks is represented as an n x n matrix where each row and column corresponds to a task. The elements of the matrix are used to indicate the presence and/or characteristics of the information flow between two tasks. Any directed graph can be converted into a DSM based representation. The simplest form of a DSM is a binary matrix that consists of 1's to represent information flow between tasks and 0's to represent that there is no information flow between tasks (see Figure 3-3(c)). It is easier to represent complex processes with greater number of tasks using the DSM. The DSM is amenable to analysis such as task sequencing and supports improvement in task sequence for increasing their concurrent execution. DSM allows representation of loops (iterations) in design processes. It is also possible to perform complex analysis on the processes by replacing the binary values in the DSM matrix to metrics that convey more information about the tasks and associated information flows. The DSM has also been used in the literature for modeling hierarchical tasks and information flows. In that case, each element in the DSM matrix corresponds to a sub-process that is associated with an independent DSM. Due to the capabilities to model processes in a systematic format amenable to analysis, the DSM approach has been employed to model design processes and is augmented to make them suitable for meta-level decisions. We discuss the existing literature on analysis and design of design processes using DSM in Section 3.3.2.

3.3.2 Design Structure Matrix for Designing Design Processes

The Design Structure Matrix (DSM) is used as an aid for meta-design. The objective is to find a sequence of tasks that converts the DSM into a lower triangular matrix, because that implies that one task can be executed after another, which in turn means that all the information required for a task is available before its execution. In complex design problem, the DSM can rarely be converted into a lower triangular form due to the coupling between tasks. Hence, the DSM is converted into a block triangular form where the number of coupled elements is as low as possible.

The DSM is mainly used for performing two types of operations on the design processes – a) partitioning, and b) tearing. *Partitioning* refers to re-sequencing the design tasks to maximize the availability of information required at each stage of the design process (Kron 1963; Gebala and Eppinger 1991). The key objective in partitioning (as used in the DSM literature) is to identify the best sequence in which design tasks should be executed such that the amount of information available for executing a design task and the concurrency between tasks are maximized. Partitioning also results in the identification of tasks that need to be executed in a coupled fashion. The partitioned design matrix is then simplified to executable form through tearing. *Tearing* refers to re-sequencing coupled tasks to find an initial ordering to start iteration. Tearing involves removal of dependence between coupled tasks and/or assumption of some piece of information required to initialize iteration. Various algorithms are developed to facilitate partitioning and tearing Design Structure Matrices.

Both partitioning and tearing are the key operations performed on DSM to modify the design processes. It is important to note that "the tool (DSM) does not actually show how to alter a design process; it merely provides a framework for analyzing alternatives"

(Eppinger 1991). The design processes can be altered in a number of ways depending on the objectives for meta-design such as maximum concurrency, minimum iteration, minimum feedback of information, minimum coupling between tasks, etc. As discussed before, the binary DSM represents only strict precedence relations. However, all the couplings between tasks are not of same strength. Some task dependencies are stronger whereas others may be weaker. The DSM is extended by including measures of degree of dependence, task durations, and execution time for different tasks. Other metrics that are considered for partitioning and tearing in DSM include task communication time, functional coupling, physical adjacency, electrical or vibration characteristics, parameter sensitivity, historical variance of task results, certainty of planning estimates, or volume of information transfer (Eppinger, Whitney et al. 1994). Multiple metrics can also be combined together to obtain design processes that satisfy multiple meta-design objectives. In the DSM, the strength of dependence is measured in terms of an importance ratio, which is determined by interviewing engineers. Statistical metrics are also developed to understand the iteration in design processes when the probability of additional iteration is known for a given sequence of interdependent tasks. Markov chain analysis is then used to determine the total iteration time. Another extension of the DSM involves quantifying the portion of information produced during the first iteration to be used during the second iteration. This metric allows analyzing the rework in a design process consisting of coupled system of tasks. Summarizing the use of DSM in designing design processes, the DSM serves as a basic framework that can be used with various metrics to configure design processes.

The DSM can be used to model relationships between *tasks* and between *parameters*. In the task level DSM, the tasks are represented in rows and columns and the elements of matrix represent information dependencies between these tasks. In the parameter level DSM, the dependencies on parameter values are modeled. This kind of formulation of DSM is useful in determining precedence relationships on the decisions about design parameters. Hybrid models that combine both task and parameter level DSMs are also developed. ***In this dissertation, we leverage the parameter level DSM to model simulation-based design processes.***

The Design Structure Matrix has been used in determining the best sequence of tasks, but *due to the inherent structure of the matrix, the DSM is limited in its application for designing design processes*. These ***limitations*** are discussed next.

1. *The DSM assumes a linear relationship between tasks and parameters*. In simulation-based parametric design, this implies that the relationship between parameters is constant, independent on the values of the parameters. However, due to the non-linearity in an actual system, the strength of relationships between parameters is generally a very strong function of the values of design parameters. The strength of relationship has an impact on the strength of coupling between design parameters. The design parameters may be coupled in one region of the design space and decoupled at other regions. Such a behavior is not captured using the matrix representation, which is good for linear transformations only.
2. Assuming that the system is linear and it can be modeled using a DSM. In such a simple case, numerical values can be used to represent coupling between tasks/parameters in the DSM. The DSM matrix can then be used to determine

strengths of coupling between sets of design parameters and the best sequence in which the parameter values can be determined. The DSM can also be used to determine the most important couplings, but the implications of a particular coupling on the designer's overall decision making power are not considered in the DSM literature. Since our focus is to design products and design processes from a decision making perspective, it is essential to quantify the impact of coupling on decision making. *The current form of DSM does not address the strength of coupling between decisions*.

3. The strength of coupling between parameters/tasks in DSM literature (Pimmler and Eppinger 1994) is measured in terms of qualitative terms such as required, desired, indifferent, undesired, and detrimental. *Required* information flow is necessary for functionality, *desired* information flow is beneficial, but not absolutely necessary for functionality, *indifferent* information exchange does not affect functionality, *undesired* and *detrimental* information exchange causes negative effects. Numerical values ranging between +2 and -2 are assigned to these qualitative terms. Such quantification is based on experience and human judgment and is sufficient (and possibly the best we can do) for measuring the coupling between tasks that involve human beings. However, in the simulation-based design, where relationships between parameters are well defined and strictly based on physics, a better estimate of the strength of coupling between parameters is conceivable. These *couplings between parameters are generally complex and hence, can't be represented simply as numbers in the DSM*.

4. The structure of *DSM does not allow capturing the impact of uncertainty* (and its propagation) due to *a)* simplification of design processes and *b)* approximate simulation models on the design decisions.
5. *Designers' Preferences are not included in designing of design processes using DSM*. This is a gap in the existing literature and is not a limitation of the DSM matrix. The set of design parameters can be extended to include additional parameters such as utility values that represent the designers' preferences. Eppinger (Eppinger 1991) pointed out that while making meta-level decisions about the design processes, the designers have to account for the tradeoff between the accuracy in design by considering coupling between design tasks/parameters and the level to which design processes can be simplified. This tradeoff has not been quantified in designing design processes using DSM. The knowledge about this tradeoff can be utilized for meta-level decision making by capturing designer's preference for the performance of design processes.

As a summary, one of the main set of limitations in designing design processes using DSM approach is due to its linearity. Another set of limitations arise from the rather simplistic set of metrics used for measuring coupling between the parameters. DSM is useful in determining the sequence of tasks but for simulation-based parametric design, it is possible to gather more information and design the design processes in a more efficient manner. Hence, DSM is used as a backbone for configuring design processes. In the next section, we discuss the details of simulation-based design processes, which is a focus of this dissertation and provide a high level view of how it is used in the method discussed in Section 3.5.

3.3.3 Simulation-based Design Processes – Networks of Information Transformations

The objective in this dissertation is to design simulation-based design processes, where networks of computer-based simulation codes are available to generate design information which is used for design decision making. The capabilities of parametric DSM are utilized in this dissertation to configure the sequence of decisions and model execution. The parametric DSM is also used to identify coupling between different decisions and models that increase the complexity of design processes. The analysis of coupling strength is however not performed in the manner proposed by Pimmler and Eppinger (Pimmler and Eppinger 1994). We use value of information based metrics to determine the impact of coupling on overall design. Before discussing how the DSM is used for initial configuration of design processes, we discuss the scope and view of design processes adopted in this dissertation.

The ***scope of design processes*** considered in this dissertation is simulation-based parametric design. Hence, the information being transformed consists of a set of parameters. The parameters are either associated with a set of values or a single point value. Design information has traditionally been categorized into three categories – form, function, and behavior. Form is described by a set of parameters that can be controlled by the designer. Form parameters have a direct impact on system's behavior. The behavior of a system can be determined directly from the system's form and the interaction with the environment. Since we are dealing with simulation-based parametric design, we assume that the conceptual design has already been performed and the concepts for satisfying functional requirements have already been selected. Our focus is on early embodiment design phase, and hence, function is not dealt with in this dissertation. The most important transformations in design processes, as highlighted by Gero (Gero 1990) include *a)* transformation of form into behavior and *b)* transformation of behavior into form. The former transformation is also called analysis, and is carried out using behavioral (simulation) models which are entirely based on the underlying physical phenomena. The latter transformation is referred to as synthesis, which is the inverse of analysis transformation. The objective during synthesis is to determine appropriate values of form parameters that satisfy given behavioral specifications. The form and behavior parameters, along with analysis and synthesis transformations can be modeled using the DSM matrix as shown in Figure 3-4. Although there are other transformations in design, we limit our discussion in this chapter to these two transformations only. The details of information transformations in design are discussed in Chapter 7. With this background of DSM for modeling design processes, we move on to a discussion of a sample design example to be used in the remaining part of the chapter.

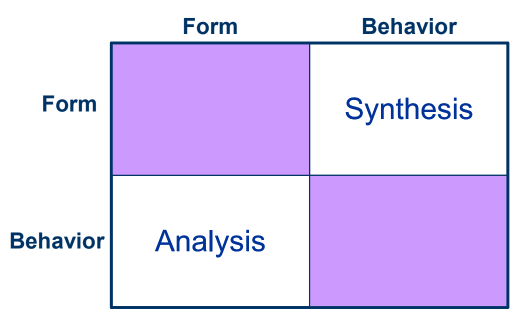


Figure 3-4 – Representation of form, behavior and associated transformations in parametric DSM

3.4 A Simple Motivational Design Example – Design of Structures

The components of the method for integrated design of products and design processes presented in this chapter are illustrated using a simple design problem – designing a structure from a given starting topology and knowledge about applied forces such that the overall weight of the structure is as low as possible. The problem is chosen because of its simplicity and the ability to directly map the structure to a corresponding decision network. The details of the problem are discussed in Section 3.4.1 and the relevance of this problem to the method discussed in this Chapter is discussed in Section 3.4.2 .

3.4.1 Structure Design Problem

The problem is adapted from Chapter 15 of the book by Vanderbei (Vanderbei 1996). In this problem, the objective is to design a structure with pre-specified loading conditions such that resulting structure has the minimum possible weight. The structure is defined by a number of nodes, which are connected by a set of members. For example, consider the structure shown in Figure 3-5. The structure has a network topology with Nodes N1, N2, N3, N4, and N5. The elements connecting the nodes are E12, E13, E14, E23, E24, E34, E35, and E45. The elements are straight lines connecting the nodes. Edge Eij connects nodes Ni and Nj, and is therefore represented as Eij = {Ni, Nj}. Eij and Eji denote the same element. The topology of the structure can be represented in the form of a matrix S , where *a)* $S_{ij} = 1$ if there is an element between node Ni and node Nj, *b)* $S_{ij} = 0$ if there is no element connecting nodes Ni and Nj. For the structure shown in Figure 3-5, the corresponding matrix is:

$$S = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

It is assumed that the forces are applied only on the nodes. In the problem, the starting configuration of the structure is known. All the elements are connected with pin joints. The constrains to be satisfied for the design problem result directly from the force balance conditions on the nodes and the moment balance conditions for the structure. The topology design problem is summarized as –

Given:
- Location of nodes,
- External Forces applied on the nodes,
- The elements connecting

Find:
- The thickness of each element

Satisfy:
- The force balance at the nodes
- Moment balance for the structure

Minimize:
- Overall weight of the structure

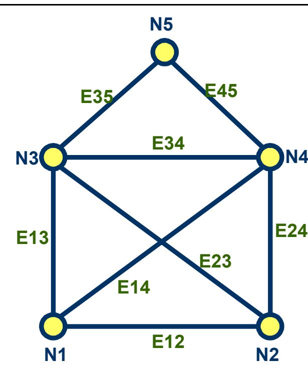


Figure 3-5 – Example topology of a structure

The force exerted by a member Eij on the end nodes (Ni, Nj) is represented by x_{ij}. If x_{ij} is positive, it represents tension in the member, whereas if it is negative, it represents compression. Unit vector along an element Eij, from node Ni to node Nj is denoted by u_{ij}. The net external force applied at a node Ni is denoted by b_i. Using the force balance conditions at a node Ni is expressed as the following linear constraint:

$$\sum_{j:S_{ij}=1} u_{ij} x_{ij} = -b_i$$

For a structure with n nodes, there are n constraints for force balance in each orthogonal direction. Hence, for a two dimensional truss, the number of constraints is twice the number of nodes. These n constraints can be written in matrix form as:

$$Ax = -b$$

The matrix A consists of the unit vectors describing the elements of the structure, and is also called the incidence matrix. The objective is to minimize the weight of the structure, which is the sum of weight of individual elements. Considering uniform material properties for all elements, the weight of each element is proportional to the product of their length and the cross-sectional area. If we assume that the stress in each of the links is same, the cross-sectional area of each element is proportional to the force acting on the link. Hence, the weight for each link is proportional to the product of its length and the force acting on it. The problem can be formulated as the following traditional optimization problem:

$$\text{Minimize } Obj = \sum_{\substack{S_{ij}=1 \\ i>j}} l_{ij} \left| x_{ij} \right|$$

$$\text{Subject to } \sum_{j:S_{ij}=1} u_{ij} x_{ij} = -b_i \qquad i = 1..n$$

A two dimensional structure is a truss if there are exactly $(2n-3)$ members. There is exactly one solution if a structure is a truss. However, if the structure has redundant members, then there may be more than one structure that satisfies the force constraints. The configuration with lowest weight will be the selected by solving the optimization problem. Please note that only structures with more number of elements than that required for a truss are interesting from the optimization perspective. There are five nodes in the structure presented in Figure 3-5. The number of elements corresponding to a truss with five elements is (2*5-3)=7. Since there are eight members in the structure in Figure 3-5, this is a redundant structure and there are multiple options that satisfy the force constraints. These options result in structures with different weights.

Note that the constraints in the optimization problem are all linear in the force acting on the members. However, the objective function is non linear due to the presence of absolute values of forces. Hence, a non-linear optimization program can be used to minimize the overall weight of the structure. The problem can also be converted to a linear optimization problem by replacing the variable x_{ij} with difference between two non-negative variables as follows:

$$x_{ij} = x_{ij}^{+} - x_{ij}^{-}, \qquad x_{ij}^{+}, x_{ij}^{-} \geq 0$$

x_{ij}^{+} is the tension part of the force in the member, and x_{ij}^{-} is the compression part. The absolute value of the force $\left| x_{ij} \right|$ is therefore equivalent to the sum of the tension part and compression part. The linear optimization problem is formulated as:

$$\text{Minimize } Obj = \sum_{\substack{S_{ij}=1 \\ i>j}} l_{ij}(x_{ij}^{+} + x_{ij}^{-})$$

$$\text{Subject to } \begin{array}{ll} \sum_{j:S_{ij}=1} u_{ij}(x_{ij}^{+} - x_{ij}^{-}) = -b_i & \qquad i = 1..n \\ x_{ij}^{+}, x_{ij}^{-} \geq 0 & \end{array}$$

A program for solving general structure optimization problems is written in Matlab. The linear optimization routine ‘linprog’ is used to optimize the objective. The program is used to determine the dimensions of starting structure shown in Figure 3-5 with loads applied to all five nodes as shown in Figure 3-6(a). The resulting structure obtained by solving the linear optimization problem is shown in Figure 3-6(b). The dotted lines are used to show where a starting element was present in the starting structure but were removed as a result of weight minimization. The line widths in Figure 3-6(b) are proportional to the forces acting on corresponding members (and hence, their cross-sectional area).

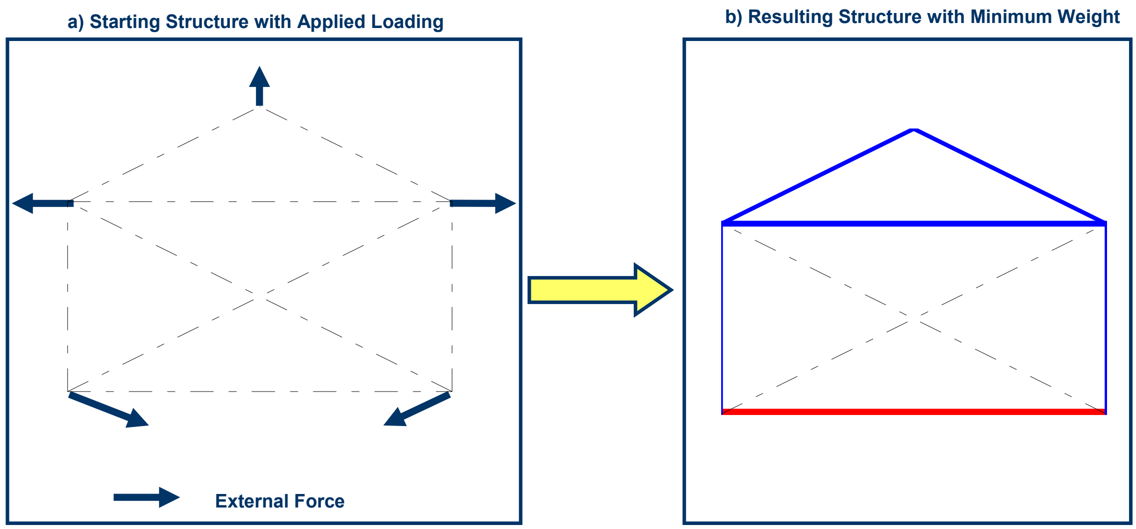


Figure 3-6 – Design of structure with five nodes and specified loading

3.4.2 Structure Design Problem Modeled as a Design Process

Although the structural design problem is interesting in its own right, the solution of the problem itself is not directly relevant to the core contribution in this dissertation. What is more interesting, is the relationship of this problem to design processes. The structure design problem involves deciding on the cross-section area of each element. There are eight decisions in the topology shown in Figure 3-5. Given the external load applied on a node, decisions related to each element connected to that node depend on each other (i.e., decisions are coupled). If we denote the decision related to element E12 as 'd1', decision related to element E13 as 'd2', and so on (see Figure 3-7(a)), we can illustrate the dependencies between decisions in the form of a network as shown in Figure

3-7(b). Therefore, a structure design problem can be viewed as a network of decisions that are related to each other.

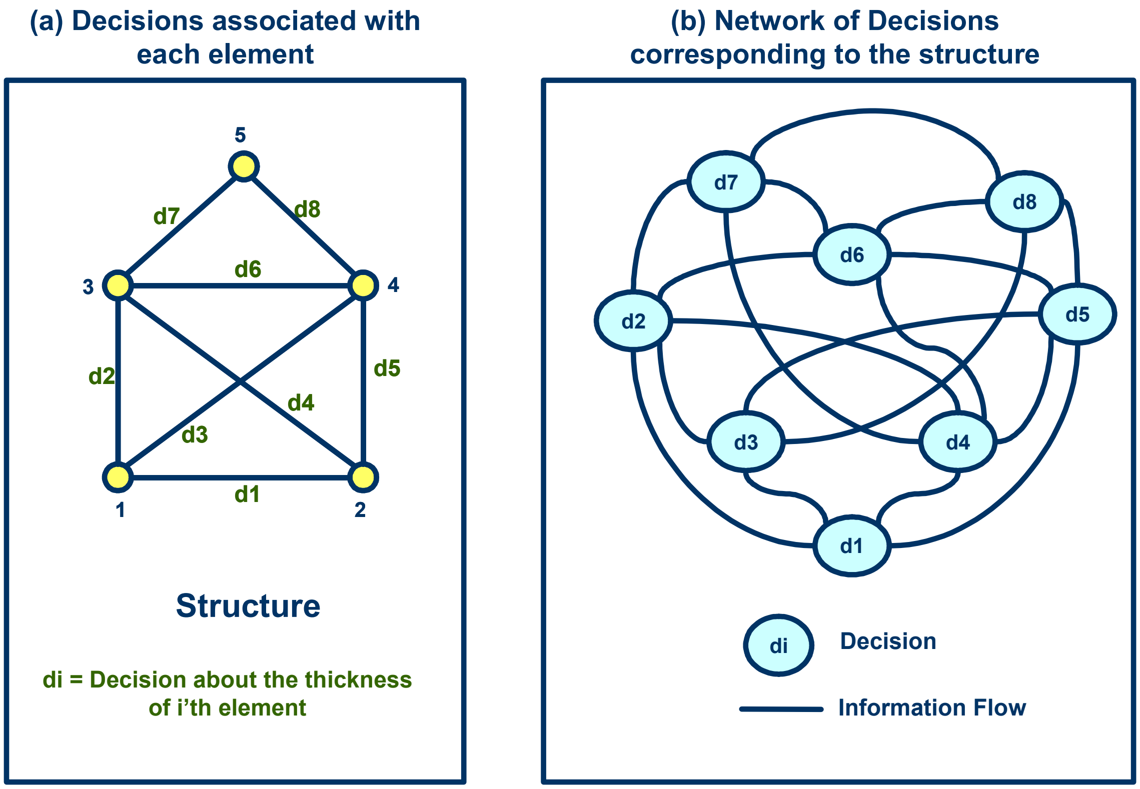


Figure 3-7 – Mapping the topology of the structure to a decision network

The network in Figure 3-7(b) is non-directional, which implies that there is no decision-making sequence of imposed on the network. Hence, a coupled decision-making is suggested from the network. From the perspective of designing corresponding elements of the structure, coupled decision making refers to the fact that the complete structure needs to be designed together. There is no subset of the structure that can be designed independent of the other part of the structure. If there is a directionality imposed on the network, it means that some of the decisions can be made in an independent manner and the result of these decisions can be used for making remaining decisions. All the decisions are linked with each other, implying that the decisions cannot be carried out in parallel either. From a meta-design (designing design process) perspective, the objective is to understand the topology of this network and the manner in which decisions are linked with each other, the strength of linkages between decisions (i.e., the information flows) etc. The network representation does not show the strength of linkage between decisions. The strength of relation between decisions for the structure design problem is directly a function of the topology and geometry – the angles between elements and the lengths of elements, which is captured in the incidence matrix A described in Section 3.4.1. The incidence matrix consists of one column for each element, which is equivalent to saying one column for each decision. There is one row for each node for each dimension. Hence, the total number of rows is equal to the product of nodes and dimensions. The numerical value of a number in the incidence matrix represents the component of unit vector in the direction of an element at a given node. The incidence matrix corresponding to structure from Figure 3-5 is shown in Table 3-2. Note that there are two rows for each node because this is a two-dimensional problem. The network representation of decisions can be directly generated from the incidence matrix.

Table 3-2 - Incidence matrix for stucture shown in Figure 3-5

	d1	d2	d3	d4	d5	d6	d7	d8
Node 1	1	0	0.70711	0	0	0	0	0
	0	1	0.70711	0	0	0	0	0
Node 2	-1	0	0	-0.70711	0	0	0	0
	0	0	0	0.70711	1	0	0	0
Node 3	0	0	0	0.70711	0	1	0.70711	0
	0	-1	0	-0.70711	0	0	0.70711	0
Node 4	0	0	-0.70711	0	0	-1	0	-0.70711
	0	0	-0.70711	0	-1	0	0	0.70711
Node 5	0	0	0	0	0	0	-0.70711	0.70711
	0	0	0	0	0	0	-0.70711	-0.70711

Having discussed the relationship between the structural design problem and the decision networks, we now discuss the key characteristics of the problem and its utilization in this chapter to explain the method for integrated design of products and design processes. The first reason for selecting this problem is that the problem is that there is a one-to-one correspondence between the topology of the structure and the coupling between decisions. It is easy to visualize the relationships between decisions directly from the structure. The second advantage of selecting this problem is that it can be formulated as a linear problem and hence, can be represented in a matrix form. The synthesis transformation – evaluation of form parameters (i.e., the cross-sectional area of elements) from the behavior (i.e., the tension and compression forces in the link) – is a linear relationship and can be easily represented in the DSM. It is also relatively easy in this problem to show the effect of decoupling different decisions and parameters. The strength of couplings between decisions can also be evaluated directly from the elements of incidence matrix.

In spite of the fact that the problem is simple it also includes the elements necessary to show the different aspects of the method. The problem is a realistic simulation-based design problem that encompasses all the three design transformations – analysis, synthesis, and preference evaluation. The problem can be formulated as a hierarchical multiscale design problem. As a summary, ***the problem is rich enough to demonstrate the different aspects of the method and is simple enough to do that easily***.

3.5 Method for Integrated Design of Multi-scale Products and Associated Design Processes

The design method proposed for multiscale design consists of six steps, as shown in Figure 3-8, and is based on the four elements discussed in Section 3.1. The method is called ***Robust Multiscale Design Exploration Method (RMS-DEM)***. The first three steps in this method constitute meta-design phase where the design process is designed. During these three steps, the first two elements -designing design processes and systems based approach for design processes are embodied. Steps 4, 5 involve execution of design processes (design phase). These steps embody the robust design principles. Step 6 relates to the refinement of simulation models and the design solution. Steps 3 and 6 of the method involve use of value of information based metric. In this section, only an overview of these two steps is provided. The details of evaluation of value of information and its use for design process simplification and refinement are described in Chapter 4 and Chapter 5. The key steps in the method are illustrated using the structure design problem from Section 3.4. In the following sections 3.5.1 through 3.5.5, the details of the steps in the method are provided.

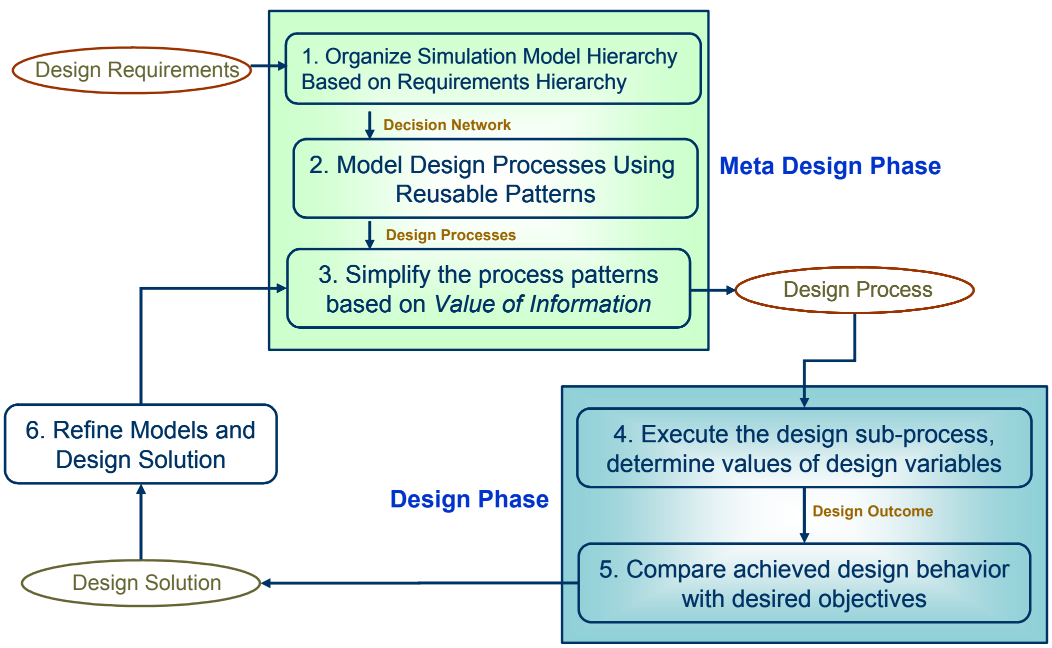


Figure 3-8 – Method for integrated design of products and design processes

3.5.1 Develop a Decision-Network Based on Available Simulation Models and Requirements Hierarchy

The first step can further be divided into the following sub-steps (see Figure 3-9):

Step 1.1. Identify the requirements/ objectives hierarchy,

Step 1.2. Assign preferences for behavior specifications

Step 1.3. Populate the available analysis models into a DSM

Step 1.4. Identify the simulation models and form attributes required to satisfy the behavior specifications

Step 1.5. Rearrange the rows and columns of DSM matrix to lower triangular matrix and identify the network of decisions (coupled and sequential)

Step 1.6. Identify analysis models that support decisions in the decision network

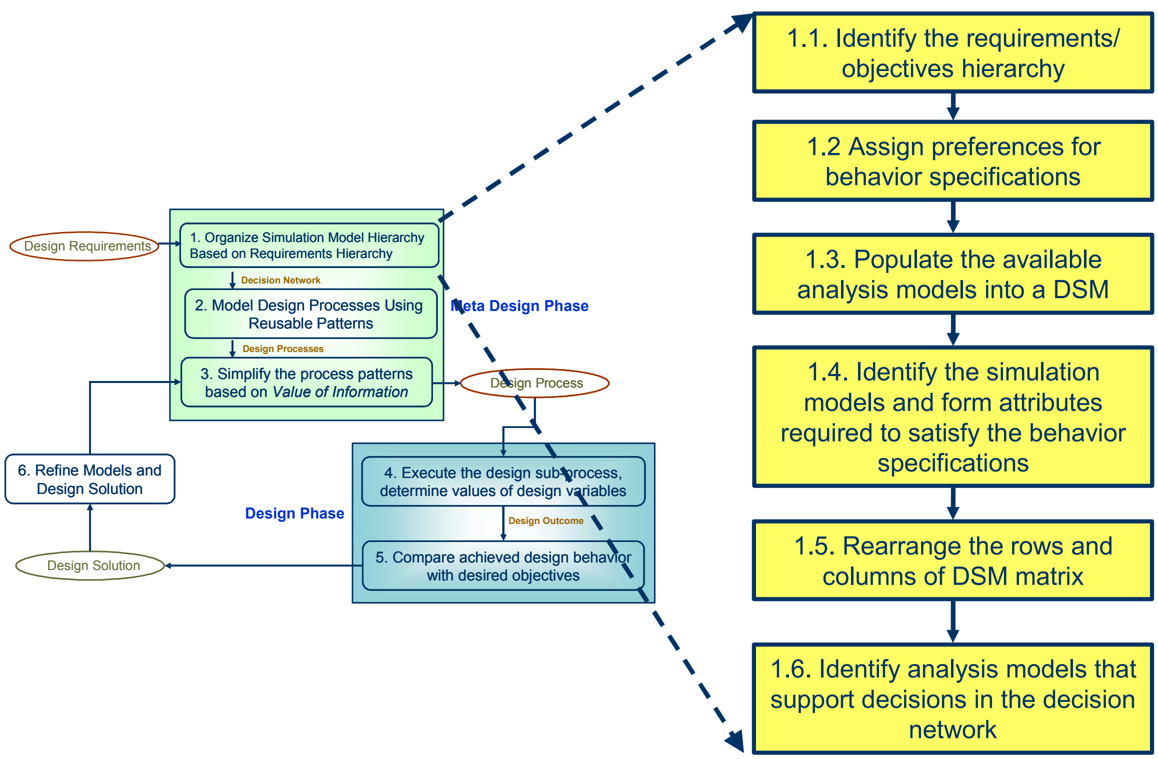


Figure 3-9 – Step 1 in the design method

Step 1.1: *Identify the requirements/ objectives hierarchy:* A design process starts with clarification of task, which results in a specification of requirements for the overall system. These requirements are broken down hierarchically into requirements for individual subsystems. Requirements for individual sub-systems are then partitioned in terms of multiple functions, associated performance characteristics, and properties. For example, during the design of automobiles, the performance requirements are defined in terms of the engine power, structural requirements, fuel consumption, etc. This hierarchical structure of requirements, performance and properties is termed as the *requirements hierarchy*. It is desired that the leaves of the requirements hierarchy be as detailed as possible. For simulation-based parametric design discussed in this dissertation, we assume that requirements can be hierarchally described to the level of parameters that describe the system behavior. The objective of design is to select appropriate values for the form variables that satisfy the target values for parameters describing system behavior.

Step 1.2: *Assign preferences for behavior specifications:* After the requirements are specified in terms of the behavior parameters (attributes), the next step is to quantify the individual preferences for each of the behavior parameters. These individual behavior specifications map different levels of achievement of an attribute to a real number. The preferences can be specified in a variety of ways such as specifying target values, value functions, etc. In this research, the preferences are quantified as utility functions. The details of utility functions and their formulation are discussed in Section 2.4. The individual preferences for behavior parameters are combined together using system level preferences that determine how much the achievement of one behavioral parameter is valued over the other. These system level preferences are useful in evaluating different design options with the tradeoffs between different conflicting conditions. Formalization of system level preferences is important for multi-functional design.

Step 1.3: *Populate the available analysis models into a DSM:* The target values of properties are achieved by appropriate values of design variables that define the *form* of the system. Having modeled the behavior attributes and the preferences, the objective is to determine the design variables (associated with the form) that can be used to satisfy the requirements and the sequence in which they must be evaluated. This is carried in the steps 1.3, 1.4 and 1.5 using the DSM matrix (Kusiak, Wang et al. 1995). The DSM captures the relationship between the requirements hierarchy and the form parameters.

It is assumed that during the start of design process, a number of simulation models are available at different scales of length and time. The input/output information for these models is used to determine the flow of information between these models. These models are then organized hierarchically based on their length scales to form a *simulation model hierarchy*. After the two hierarchies for decisions and simulation models are developed, the next step is to map the two hierarchies in order to determine which simulation models can be used to support different decisions. Some of the decisions can be made in parallel, while others need to be made in a sequential fashion. The objective of this step in the method is to organize the simulation models by imposing precedence relationships, the result being a sequence (network) in which the models need to be executed in order to satisfy the design requirements.

Assuming that the designers have at their disposal a list of analysis models with some parameters as inputs and outputs, a parametric DSM is formulated. The rows and columns of the DSM correspond to the parameters that are inputs/outputs in the analysis models. An element D(*i, j)* of the DSM matrix is populated with the corresponding analysis model if there is an analysis model that has j^{th} parameter as input and the i^{th}

element as an input. This is illustrated using an example in Figure 3-10. In this figure, two analysis models – A1 and A2 are shown along with five parameters a, b, c, d, and e. The input and output parameters for both the analysis models are shown in the figure. Corresponding to these inputs and outputs, a parametric DSM matrix is also shown, which is used in Steps 1.4, 1.5 and 1.6. The matrix is useful in determining the coupled parameters. For example in Figure 3-10, the analysis models A1 and A2 are coupled because there is an element above the diagonal in the matrix. The DSM matrix can be partitioned into analysis and synthesis blocks by identifying the parameters as form parameters and behavior parameters. The variables that are neither form variables nor behavior variables are intermediate variables. In the example, if a and b are form variables, d and e are behavior variables, c is an intermediate variable. Submatrices that are associated with analysis and synthesis can be identified from the complete DSM matrix by identifying the design variables and response variables. The DSM matrix is organized as shown in Figure 3-4.

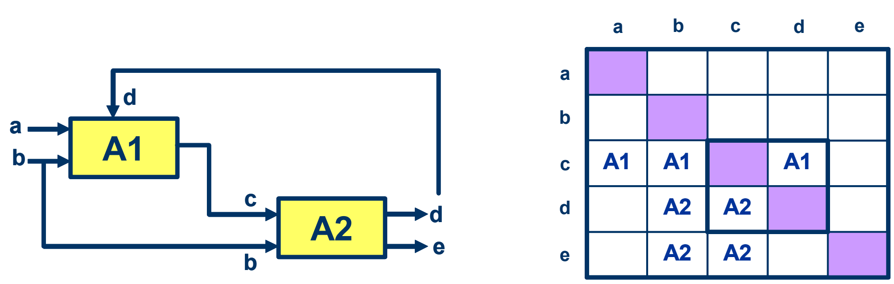


Figure 3-10 – Example showing use of parametric DSM for modeling network of analysis models

Step 1.4: *Identify the simulation models and form attributes required to satisfy the behavior specifications:* Using the behavior parameters identified from requirements in Step 1.1, the DSM matrix is traced to identify the minimum set of design variables that can be used to satisfy the design requirements. The parameters that affect a given behavior parameter can be identified by traversing the rows corresponding to the behavior specification and determining the columns (and corresponding design variables) that have a non-empty element in the DSM. The analysis model marked in that element of the DSM matrix is required for predicting the behavior. For example, in the DSM, the non-empty elements corresponding to row e are b and c. Hence, in order to evaluate the behavior parameter e, the required inputs are b and c. Similarly, the parameters and models required to evaluate these parameters can be determined by traversing the rows corresponding to b and c. After all the required form and intermediate parameters are determined, the remaining parameters can be eliminated from the DSM matrix.

Step 1.5: *Rearrange the rows and columns of DSM matrix to lower triangular matrix and identify the network of decisions (coupled and sequential):* In this step, the DSM that consists of the minimum required parameters is rearranged to maximize the amount of information available before executing a simulation model. This is important to maximize the concurrency in executing simulation models and choosing the right sequence for execution of analysis models. Algorithms for performing this step are available in the DSM literature. This step is also important because it helps in identifying the couplings between simulation models that need to considered, and the sequence of those couplings. In general, the impact of a coupling that exists earlier in the analysis chain is greater than a coupling later in the analysis chain. Using this re-organized DSM matrix, the network (sequence if the design variables are not coupled) in which the values of design variables need to be selected is determined. This network is referred to as the decision network.

Step 1.6: *Identify analysis models that support decisions in the decision network:* The decisions in the decision network can then be assigned to analysis models that generate information for decision making. This step is performed by identifying the analysis models from the elements in the DSM matrix. This concludes the first step in the design method. The outcome of the Step 1 in the design method is formulation of decision networks with corresponding simulation (analysis) models required to generate information for executing those decisions. The second step in the design method is to identify repeating patterns of decisions and models and assignment of design processes to those patterns. The second step in the method is discussed in detail in Section 3.5.2.

3.5.2 Model Design Processes Using Reusable Process Patterns

The second step in the design method involves developing the network of decisions further into a design process at the level of detail consisting of tasks and information flows. This step is carried out by recognizing that the design process can be broken down into standardized building blocks. The building blocks are considered to be standard because not do only they occur in any design process, but also they can be used to model any design process. The building blocks used for modeling design processes are identified by observing regularities in design processes. According to Nikos A. Salingaros (Salingaros 2000), "the ability to observe patterns gives us the human advantage of both adapting to, and changing our environment". Patterns represent regularities that recur in a particular design domain, and have been successfully used in architecture and design of software programs. According to Alexander and co-authors (Alexander, Ishikawa et al. 1977), a pattern describes a problem that occurs over and over again in our environment, and then describes the core of solution to a problem, in such a way that the solution can be used a million times. Patterns can be defined in many different ways – in software design, Gamma and co-authors (Gamma, Helm et al. 2000) describe patterns in terms of the behavior of objects, structure of interactions between objects, and the manner in which they are created. Alexander describes patterns for architectural design in terms of neighborhood boundary, main gateways, arcades, etc.

It is important to note that there is no unique way of describing the patterns. Patterns are generally identified by recognizing certain characteristics of the system that are important in a given context. Since the context in this dissertation is simulation-based multiscale design, we have identified patterns based on interactions between simulation models at multiple scales and between decisions. Interaction between decisions and models is taken as a basis for defining patterns in design processes because *a)* interactions primarily define the flow of information between tasks, and *b)* the types of interaction dictate the kind of design process to be used for design. The patterns based on interactions between decisions and models are also useful in simplifying the design processes in the third step of the design method. The simplification of design processes using the interaction patterns is discussed in Section 3.5.3.

The interaction patterns used as patterns in the design method are shown in Figure 3-11 and organized in a 3x3 matrix form. The three columns of the matrix represent three different types of interactions – *i)* independent, *ii)* dependent, and *iii)* coupled. In the independent scenario, both the microscale and macroscale simulation models can be executed in a parallel fashion. The dependent scenario represents one way (sequential) flow of information, where the information generated by microscale model is fed into the macroscale model. In the coupled scenario, both these models need to be executed together with a two-way flow of information between them. Such a classification is also common the DSM literature (Eppinger 1991), where the flow of information between tasks is defined as independent tasks, dependent tasks, and interdependent tasks respectively.

The classification in three rows of the matrix is based on design variables and responses associated with different models and multi-functionality. In the first row, the macroscale model has design variables and response variables associated to it, whereas in the second row, both the microscale model and macroscale models are associated with design and response variables. The third row represents a multifunctional design scenario where at each level, there are different models that predict the system behavior for different functional characteristics (such as thermal, impact, vibration, etc.).

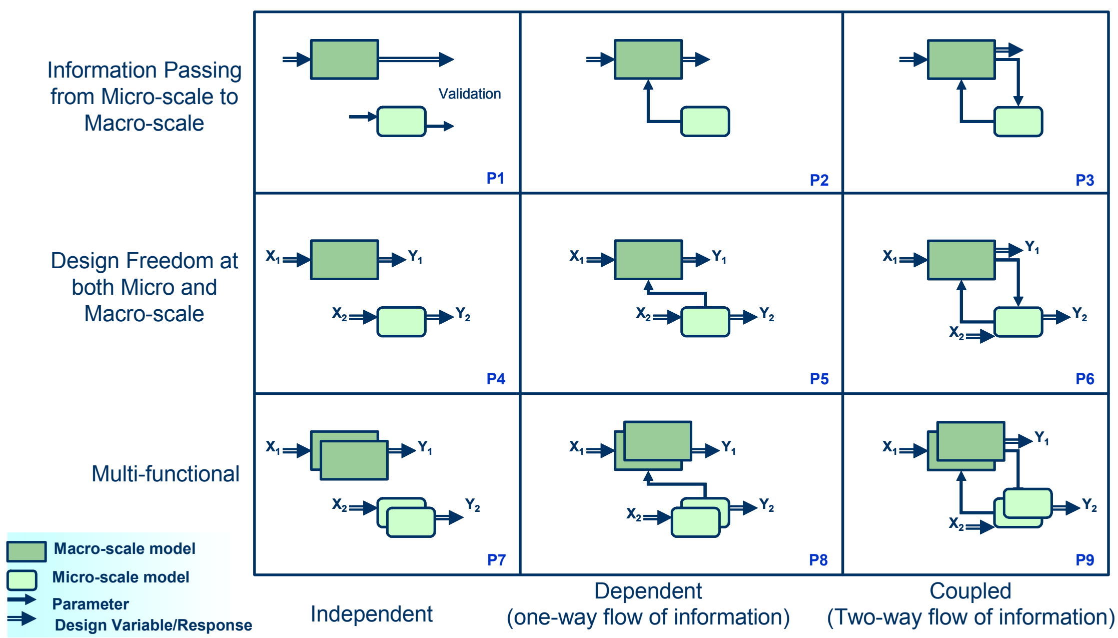


Figure 3-11 – Interaction patterns in multiscale design

Interaction patterns P1, P2, and P3 shown in Figure 3-11 are represented in a matrix form as shown in Figure 3-12. The variables labeled x_i represent design variables and the variables labeled y_i represent response variables. The boxes labeled A_i are the analysis models that transform the design variables into response variables. Similarly, the matrix representations corresponding to patterns P4, P5, and P6 are shown in Figure 3-13.

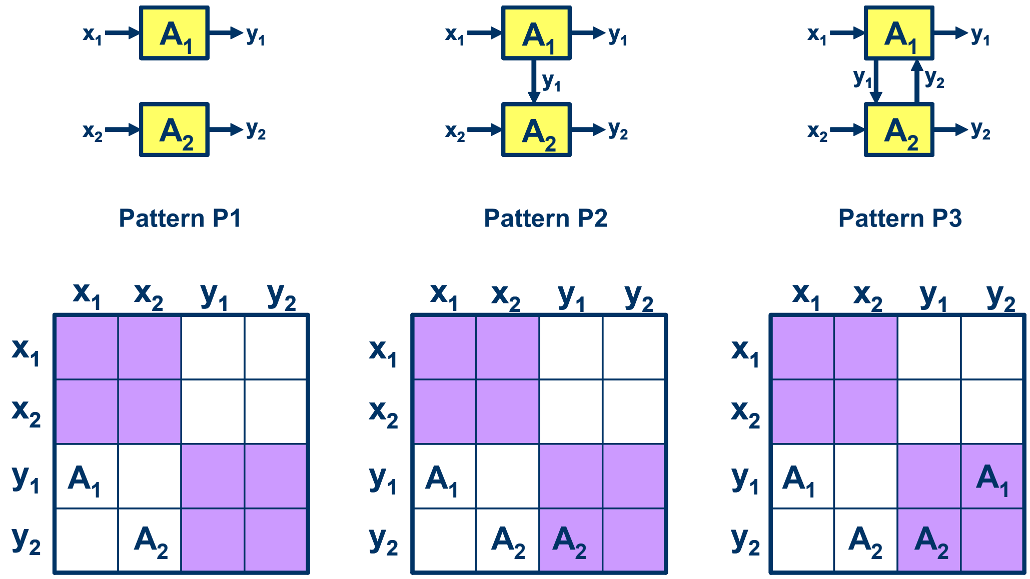


Figure 3-12 – Patterns P1, P2, and P3 in represented in the matrix form

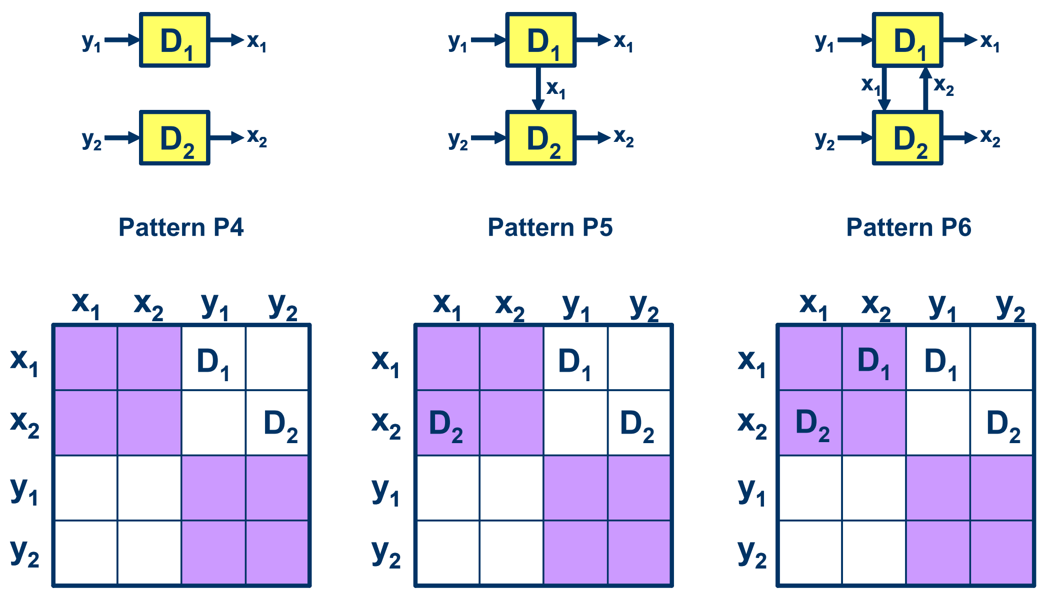


Figure 3-13 – Patterns P4, P5, and P6 represented in matrix format

The boxes labeled D_i are decisions where design variables are evaluated based on the desired response values. Notice that in both the figures, the analysis and synthesis components of the DSM matrix are labeled with corresponding analysis models and decisions. The two shaded regions in the DSM matrix correspond to the information flow between analysis models and decisions. The bottom-right shaded sub-matrix corresponds to information flow between analysis models and the top-left sub-matrix corresponds to the information flow between decisions. If the shaded sub-matrices are empty, there is no information flow (e.g., in patterns P1, P4). If the sub-matrix is lower-triangular, the information flow is sequential (e.g., in patterns P2, P5), whereas if the matrix is not lower triangular, the flow is coupled (e.g., in patterns P3, P6).

Although the DSM matrix corresponding to the interaction patterns are shown with single inputs x_i and single outputs y_i from the models and decisions, the same matrix representation can be applied for models where there are multiple inputs and outputs. In that case, the X_i and Y_i represent an array of inputs and outputs respectively. The same set of interaction patterns can be extended to represent complex decision and model networks if each model/decision in the interaction patterns represents a network of other models/decisions viewed as a black box. The black box can then be represented as interactions between other lower level black boxes. It can be shown that any DSM matrix can be represented hierarchically in terms of these patterns P1 through P9. Hence, these nine patterns are termed as the basic design process patterns. Examples of decision networks with multiple decisions and simulation models represented in terms of the basic patterns are provided in Chapter 9. The applicability of these basic patterns to any network of decisions/models is very important because it allows us to consider only these patterns for analysis of design processes and extend the results to other composite patterns that can be developed by combining the basic patterns.

Using the network of models and decisions developed as a result of Step 1, the second step in the design method shown in Figure 3-8 is identification of interaction patterns from the decision network and associated design processes. This is carried out in the following sub-steps:

Step 2.1. Model the network of decisions and associated simulation models using basic interaction patterns,

Step 2.2. Associate individual interaction patterns with design sub-processes,

Step 2.3. Compose design sub-processes to develop overall design process

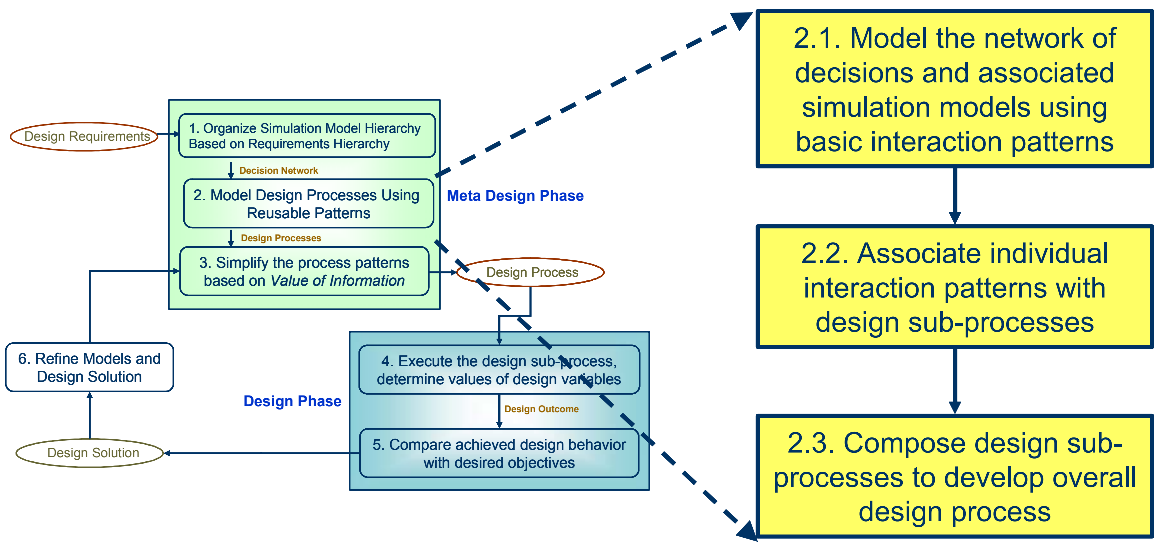


Figure 3-14 – Step 2 in the design method

Step 2.1: *Model the network of decisions and associated simulation models using basic interaction patterns*: The decision network, which is an outcome of the Step 1 is successively partitioned into two sets of models/decisions and the interaction between the two sets is identified. The two subsets can be identified by first considering an independent interaction pattern, then a sequential pattern, and finally a coupled interaction pattern. The DSM matrix can be converted into an independent pattern if it can be converted into a block diagonal form. The matrix can be converted into a sequential pattern if it can be converted into a diagonal form with block diagonals. Finally, if the matrix cannot be converted into block diagonal or lower triangular with block diagonals, a coupled interaction pattern is used. Based on this interaction, interaction patterns (P1 through P9) are assigned to the two sets. Then, each subset in the interaction pattern identified and the associated sub-matrix is considered for mapping to interaction patterns. This process is carried out until the subsets cannot be divided further. The interaction patters at various levels in the hierarchy are then assigned to design processes assigned to individual design processes in Step 2.2 and the design processes are composed together in Step 2.3.

Step 2.2: *Associate individual interaction patterns with design sub-processes:* As mentioned before, the interaction patterns are provided labels from P1 through P9. The primary advantage of this classification is that each type of interaction pattern is associated with a design process that represents the design exploration and decision making loop to be used in the overall system design process. Some examples of specific design processes associated with these interaction patterns are presented in Section 3.5.4. These interaction patterns embody the systems view of design processes because design processes can be modeled at various levels of abstraction by using the same set of patterns. In other words, the system level design processes can be *composed* using predefined design processes associated with each of these standardized interface patterns. Further, these interaction patterns are domain independent and hence, can be applied to any kind of a multiscale design problem.

Step 2.3: *Compose design sub-processes to develop overall design process:* After the design processes are assigned to individual interaction patterns, the design processes are combined together and the information flow for the complete design process is determined. The composition of design processes and assignment of design processes to interaction patterns is simple from the point of view of determining the flow of information between tasks, but it imposes a number of requirements for the computational frameworks on which the design method is implemented. The details of requirements for the implementation of this method on a computational framework are discussed in Chapter 7 and a proposed implementation is discussed in Chapter 8.

3.5.3 Design Process Simplification

Multiscale, multifunctional design processes are generally complex because of the inherent coupling between various scales. An independent type of design process takes less time to execute when compared to a decoupled (dependent) design process, which in turn takes less time than its coupled counterpart. By de-coupling a coupled system, designers can reduce the complexity of design processes but increase the uncertainty in the design. Hence, the designers are faced with the meta-level decision involving tradeoff between simplification of design process and effectiveness of the final design. Appropriate simplification of the design process by reducing coupling is very important for design exploration. In the context of interaction patters discussed in Section 3.5.2, the complexity increases from left-to-right and from top-to-bottom in the matrix of interaction patterns, as shown in Figure 3-15. Interaction Pattern P9 results in the most complex design process whereas the processes associated with Pattern P1 are the simplest. While going from the left-to-right column, the complexity increases because of increased coupling. The complexity in second row is higher than the first row because design exploration needs to be carried out at both scales. The increase in complexity from second to third row is because of the additional coupling between functions at a given scale in a multifunctional scenario. Hence, the objective during process simplification is to systematically go from Pattern P9 to Pattern P1 as shown by the circular arrows. This objective gives rise to the need for defining a metric that guides designers to determine the right level of simplification of interaction patterns. Note that in this dissertation, the scope of simplification of design processes is limited to simplification of interaction patters.

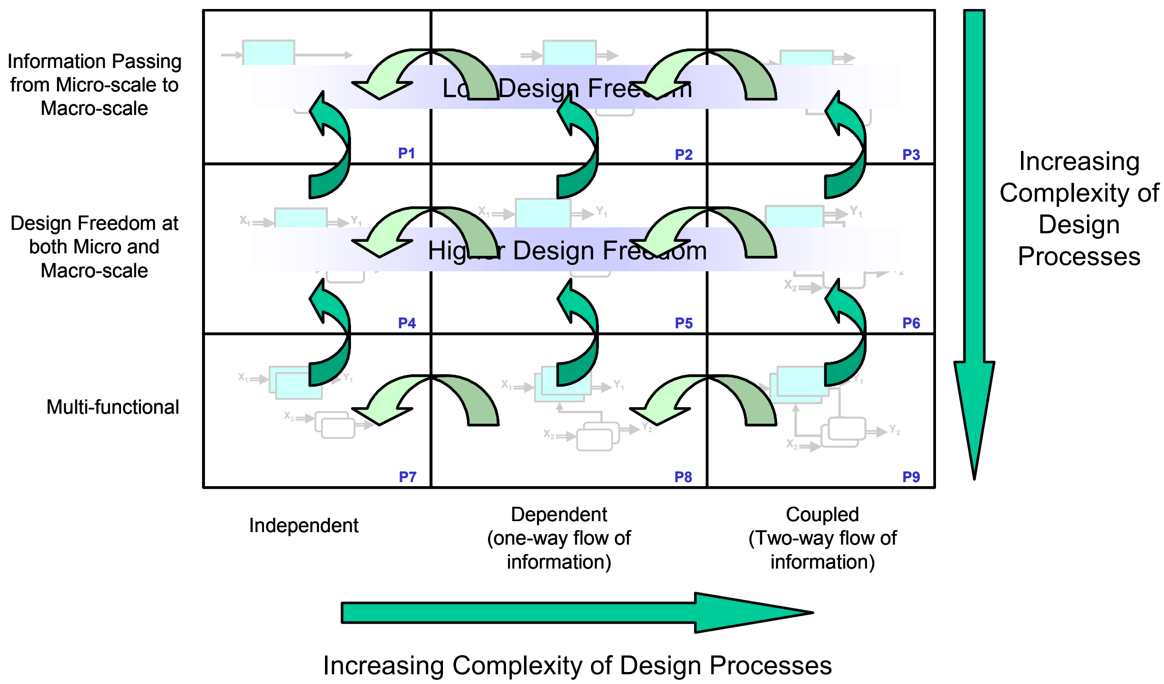


Figure 3-15 – Complexity in the simulation model interaction patterns

Related work on measuring the complexity of processes (Braha and Maimon 1998; Buede 2000) is focused mainly on the quantifying the interactions between different systems. Some of the metrics capture only the number of interacting tasks, whereas other metrics capture the strength of these interactions. In other words, existing measures of quantify the coupling between system's inherent behavior only. However, the simplification of design processes is dependent on three factors – *a)* the coupling inherent in the system behavior, *b)* the designers' preferences, and *c)* the stage in the design process. Coupling in system's behavior is captured in the simulation models and the relationship between design variables and responses. Designer's preferences may either amplify or diminish the effect of coupling in the system. Similarly, the stage in the design process may dictate whether some coupling between design variables and responses is important. For example, a simple design process may be good enough in the preliminary design phases where the objective is to reduce the number of options to a few promising options. However, in the detailed design phase, it may not be appropriate to simplify the design process. This also indicates the appropriateness of simplification is dependent on the design timeline. Designers' preferences and stage in the design process are not considered in the design literature to determine whether a coupling between design variables and response is important or not.

In this dissertation, we overcome this limitation by developing a metric that considers both system coupling and designer's preferences for determining whether a design process simplification is appropriate. A decision making perspective is adopted for making the meta-level decision under consideration. The guiding principle used for determining whether process simplification is appropriate is the answer to the following question – *"What is the impact of process simplification on the design decisions?"* If the impact on the decision is small and process simplification reduces the design exploration cost drastically, then the designer should go ahead and simplify the design process, otherwise not. In order to quantify the impact on decision, metrics based on value of information are currently being developed. Value of information refers to the benefit of additional information due to preserved couplings per unit cost of computation and extended design time. *Expected value of information*, as defined by Howard (Howard 1966), and later applied to catalog selection problems in engineering design by Bradley and Agogino (Bradley and Agogino 1994), is given by the difference between the expected value of the option selected with the benefit of information less than without. Comparing two patterns (see Figure 3-11) such as P2 (sequential interaction) and P3 (coupled interaction), by including the coupling in P3, we are adding information about the system that is not accounted for in P2. If the expected value of this added information is greater than certain threshold value, pattern P3 should be used instead of P2. Similar notion is used for scale decoupling, decision decoupling, and functional decoupling. The details of the Value of Information metric for quantifying the impact of a simplification on the designer's decision making capability is discussed in Chapter 4 and its utilization for design process simplification for scale and decision decoupling is discussed in Chapter 5.

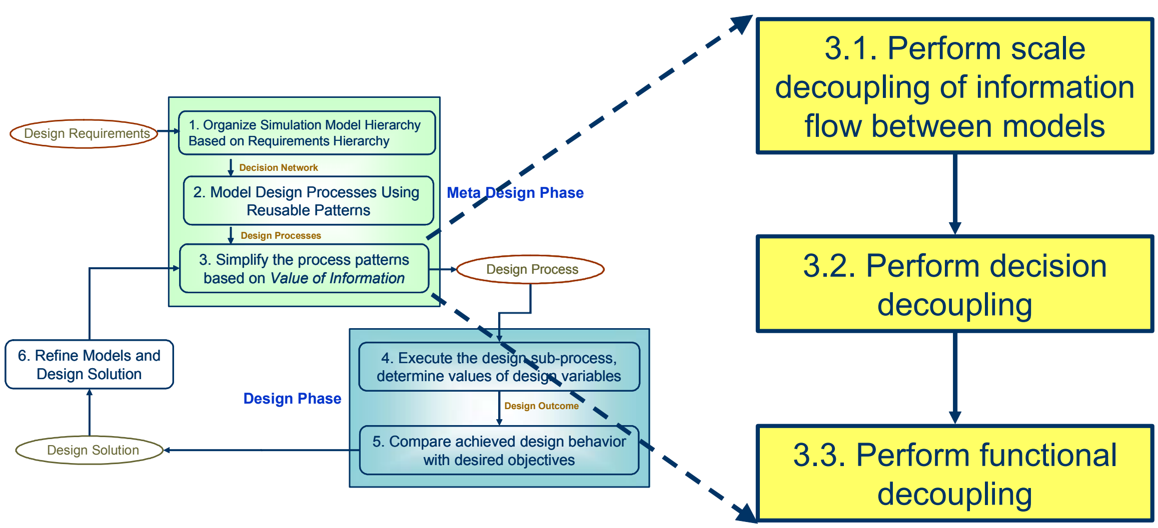


Figure 3-16 – Step 3 in the design method

Using the value of information, design process simplification in the design method is carried out through following three sub-steps –

Step 3.1. Perform scale decoupling of information flow between models,

Step 3.2. Perform decision decoupling, *Step 3.3.* Perform functional decoupling

The details of these sub-steps are discussed next.

Step 3.1: *Perform scale decoupling of information flow between models:* Scale decoupling refers to simplification of pattern P3 to P2, and pattern P2 to P1. In scale decoupling, a single design decision needs to be made. Two simulation models are available at designer's disposal for making the decision. Although the simulation models are coupled with each other, the designers may choose to simplify the decision-making by decoupling these models. This simplification would result in some error in the predicted system behavior. If the upper bound on the error of system's behavior prediction is known, the value of information for the decision can be evaluated. Based on the value of information, the designer can make the meta-level decision – "should the interaction pattern P3 be simplified to P2 or P1?" The details of scale decoupling are discussed in Section 5.3.1, and validated using a data center cooling system design in Section 5.3.2.

Step 3.2: *Perform decision decoupling:* Decision decoupling means simplification of pattern P6 to P5, and pattern P5 to P4. In this case, the designers are required to make two decisions about the form variables. The decisions are coupled with each other (pattern P6). The meta-level decision that the designer need to make is: "should the decisions be made in a coupled fashion or can the interaction between them be simplified to sequential or independent interactions?" In order to make this meta-level decision, the designer evaluates the increase in the value of information with coupling as compared to without. The information required to perform the value of information based calculation is the lower and upper bounds on design variables. The details of decision decoupling are discussed in Section 5.4.1, and validated using the datacenter cooling example in Section 5.4.2.

Step 3.3: *Perform functional decoupling:* Functional decoupling is a term for converting a multi-functional problem into a mono-functional problem (if the problem permits). Functional decoupling can be achieved by simplifying pattern P9 to P6, pattern P8 to P5, and pattern P7 to P4. Functional decoupling can be performed using the value of information metric in a manner similar to model and decision decoupling. Hence, decision decoupling has not been discussed in detail in this dissertation. The scenario that is more common and is important for multifunctional design is the one where the functional requirements are coupled with other. Chapter 6 is dedicated to design when the functional characteristics of the systems are coupled. In that chapter, a set-based focalization method is presented, where the design space is divided between different designers and portions of design space are systematically eliminated until the designers converge to a single point. The method is illustrated using a simple set of quadratic responses and validated using a Linear Cellular Alloy design problem.

Decision Decoupling in Structure Design Example

The simplification of design processes is illustrated using a structure design problem. The starting structure and the applied forces are shown in Figure 3-17. The decision network and the incidence matrix for the structure are shown in Figure 3-18 and Table 3-3 respectively. Solution of the decisions in a coupled fashion results in the overall objective function value equal to 32.515.

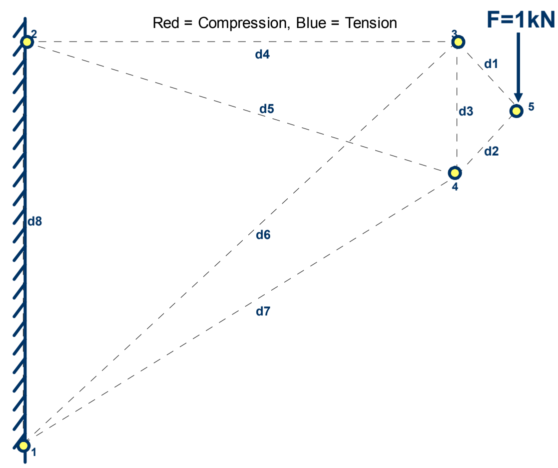


Figure 3-17 – Example truss problem to illustrate design process simplification

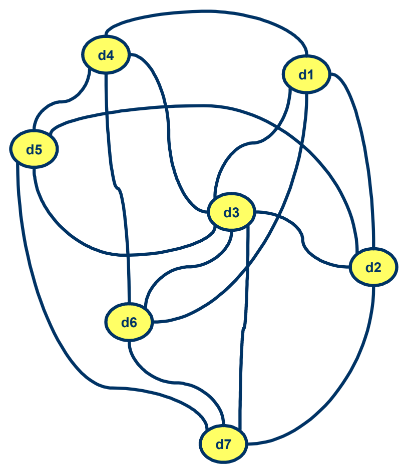


Figure 3-18 - Network of decisions for sample problem shown in Figure 3-17

Table 3-3 - Incidence matrix for the sample structure shown in Figure 3-17

	Elements							
Nodes	1	2	3	4	5	6	7	8
	0	0.89443	0.94868	0	0	0	0	0
1	1	0.44721	0.31623	0	0	0	0	0
	0	0	0	1	0.98639	0	0	0
2	-1	0	0	0	-0.1644	0	0	0
	0	-0.89443	0	-1	0	0	0.86602	0
3	0	-0.44721	0	0	0	-1	-0.50001	0
	0	0	-0.94868	0	-0.98639	0	0	0.86602
4	0	0	-0.31623	0	0.1644	1	0	0.50001
	0	0	0	0	0	0	-0.86602	-0.86602
5	0	0	0	0	0	0	0.50001	-0.50001

The problem can ideally be solved in a coupled fashion where the decisions about each of the elements can be made in a linked fashion. This refers to the coupled decision pattern P6. The interaction pattern can however be simplified by identifying that the objective function (minimization of volume/weight) is a sum of cross-sectional areas of each element, weighted by their lengths. The cross-sectional areas are directly proportional to the forces in the elements and the lengths are prespecified constants. Due to this linearity in the objective function, if a subset of decision variables can be determined independent of other design variables, then the corresponding contribution to the objective function can also be minimized independently. In other words, if the forces on a subset of elements are independent of other elements, then the weight of that subset of elements can also be minimized independently. The dependence between subsets of elements can be identified in the incidence matrix. Each row in the incidence matrix corresponds to the force balance at one node (in one direction) in the structure and each column corresponds to the elements. Two non-zero elements in a row in the incidence matrix represent coupling between the elements corresponding to those columns. The rows corresponding to node 5 in the incidence matrix shown in Table 3-3 has all zero elements except elements 7 and 8. Hence, decisions about elements 7 and 8 can be taken independent of the other elements. After the decisions about 7 and 8 are made, the decisions about remaining links can be made in a coupled fashion. This sequence of subsets of decisions is shown in the incidence matrix in Table 3-4. The corresponding structure and network are shown in Figure 3-19. The decision making pattern in this case is pattern P5. Solution of the decisions in a sequential fashion results in the overall objective function value of 32.515, which is same as the coupled case.

Table 3-4 - Decomposition of sample structure design problem into sequential decisions

	Elements							
Nodes	1	2	3	4	5	6	7	8
	0	0.89443	0.94868	0	0	0	0	0
1	1	0.44721	0.31623	0	0	0	0	0
	0	0	0	1	0.98639	0	0	0
2	-1	0	0	0	-0.1644	0	0	0
	0	-0.89443	0	-1	0	0	.86602	0
3	0	-0.44721	0	0	0	-1	-0.50001	0
	0	0	-0.94868	0	-0.98639	0		0.86602
4	0	0	-0.31623	0	0.1644	1		0.50001
	0	0	0	0	0	0	-0.8660	-0.86602
5	0	0	0	0	0	0	0.50001	-0.50001

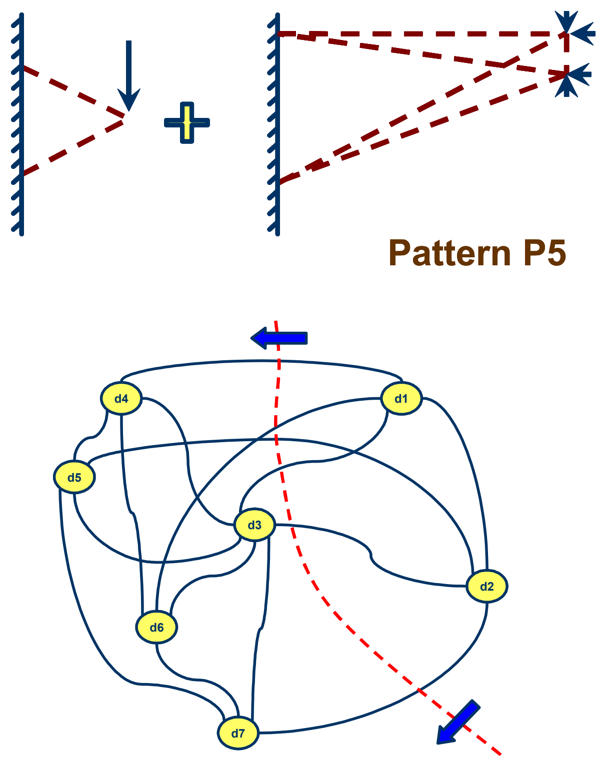


Figure 3-19 – Structure and network representation of sequential decisions in structure design problem

Further simplification of the structure is not possible because all other decisions are coupled. This is generally the case in multifunctional design scenarios. The decisions cannot be directly decoupled. However, as designers, we are not interested in finding just the optimum solutions. We are interested in satisficing solutions that are not optimum but are close to the optimum. If the design process can be simplified significantly without having a major impact on the final design, the simplification of processes is preferred. For example, in the case of the structure design problem under consideration, if we eliminate element 6 from the starting structure, then the remaining elements (1, 2, 3, 4, and 5) can be divided into two subsets - *a)* elements 2, 4 and *b)* elements 3, 5 whose decisions can be made in a parallel fashion. The removal of element 6 from the incidence matrix is represented by a X in the sixth column in Table 3-5.

Table 3-5 - Sequential and independent decision making by ignoring element 6 in the starting structure

	Elements							
Nodes	1	2	4	3	5	6	7	8
1	0	0.89443	0	0.94868	0		0	0
	1	0.44721	0	0.31623	0		0	0
2	[illegible]	0	1	0	0.98639		0	0
	-1	0	0	0	-0.1644		0	0
3	0	-0.89443	-1	0	0	X	0.86602	0
	0	-0.44721	0	0	0		-0.50001	0
4	0	0	0	-0.94868	-0.98639		[illegible]	0.86602
	0	0	0	-0.31623	0.1644		[illegible]	0.50001
5	0	0	0	0	0		-0.866	-0.86602
	0	0	0	0	0		0.50001	-0.50001

The arrows represent the sequence in which decisions are made. The structure and the associated network are shown in Figure 3-20. The decisions in this scenario correspond to interaction pattern P4 as shown in the figure. The solution of decisions in an independent fashion after removal of element 6 from the structure results in an overall objective function value equal to 33.182, which is higher than the coupled and sequential scenarios. The overall objective function (weight) is compared for the three scenarios (i.e., three decision patterns) in Table 3-6. The value is higher in the case of coupled decision because of the assumption that link 6 does not exist. This assumption simplifies the design process at the expense of the quality of decision. If the increase in the objective function value (33.182-32.515=0.667) is not significant from the design requirements standpoint, then the simplification of design process is appropriate.

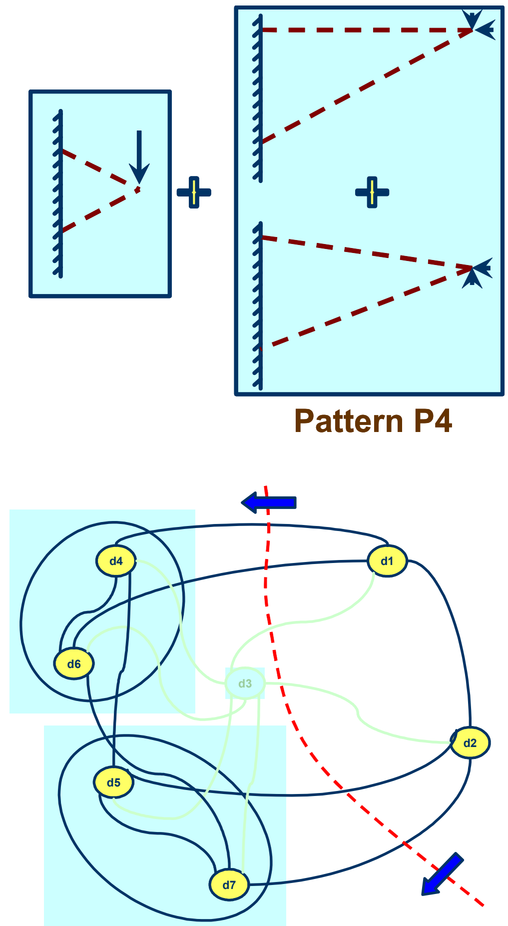


Figure 3-20 - Decomposition of decisions into sequential and independent decisions

In the previous paragraph, the impact of removing one of the elements on both the design process and the decision are discussed. It is observed that by removing the element and decoupling the design process, the process is simplified with only a minor impact on the final decision. For designing design processes, we are interested in identifying and taking advantage of such scenarios to simplify design processes. Although the design processes can be simplified in a variety of ways, our focus in this dissertation is on simplification of interaction patterns between models, decision, and functionalities. The question then is - which decisions should be decoupled so that that impact on design processes is large but the impact on final decisions is small? In the context of the structure design example, which elements should be removed from the starting structure? The obvious answer to this question is that the couplings should be removed if the tradeoff between simplification of design process and the reduction in quality of the final design is favorable.

Table 3-6 - Comparison of overall objective function value for coupled, sequential, and independent decision scenario

Pattern	*Weight*
Coupled	32.515
Sequential	32.515
Sequential + Independent	33.182

The impact of removing an element from the initial structure changes the forces on the other elements. Hence, the weight of the final structure is also different. This weight of the final structure is always greater than or equal to the structure designed without removing an element. One way to determine the impact of removing an element on the change in weight is to solve the two decision problems with and without the element (that is to be removed) and then subtract the two weights. In a general design scenario, this is equivalent to executing the complex and simplified design processes and then comparing the results. Although this would provide an accurate estimate of the impact on decision making, it defeats the purpose of designing design processes. Determining whether a simplification is appropriate should not require designers to execute the complex design process because in that case, there is no need for simplification. We already have an accurate design solution. This is one of the key challenges in designing design processes.

We want to make decisions about the design processes without executing the design processes themselves. In order to address this challenge, designers should not try to come up with exact estimate of the impact on their decisions. Rather, they should use some indicators or metrics that are related to the impact of simplification on decisions and can separate appropriate simplifications from inappropriate simplifications. For example, since the structure design problem is formulated as a linear programming problem, the feasible design space is bounded by a convex region and the optimum lies at the intersection of the constraints. With this knowledge, the maximum possible value of any design variable (forces in the elements) can be calculated fairly easily. This maximum possible value of force in an element multiplied by the length of the element is equal to the upper bound on the impact on objective function value. Hence, without calculating the forces in the individual elements, the upper bound can be calculated. If the upper-bound of impact on the objective function is less than some prespecified value, then the designers can simplify the associated design process without reducing the quality of the decisions. In this dissertation, we use a similar metric for determining the impact of a simplification on the designers' decisions. This metric is discussed in Chapter 4. Based on this metric, methods are developed for systematic simplification of processes in Chapter 5.

The outcome of design process simplification (Step 3 in Figure 3-8) is a *design process* that is executed to result in the values of design parameters for satisfying design requirements. This finishes the meta-design phase.

3.5.4 Design Process Execution and Design Verification

The design process developed in the meta-design phase is then formulated in such a manner that it can be executed. In this step of the design method, elements of the DSP Technique discussed in Section 2.2 are leveraged. The step involves formulation of design decisions as support problems and their execution. The step consists of the following substeps:

Step 4.1: Formulate the decisions as decision support problems

Step 4.2: Assign design processes for execution of support problems

Step 4.3: Execute design processes to determine the values of design variables

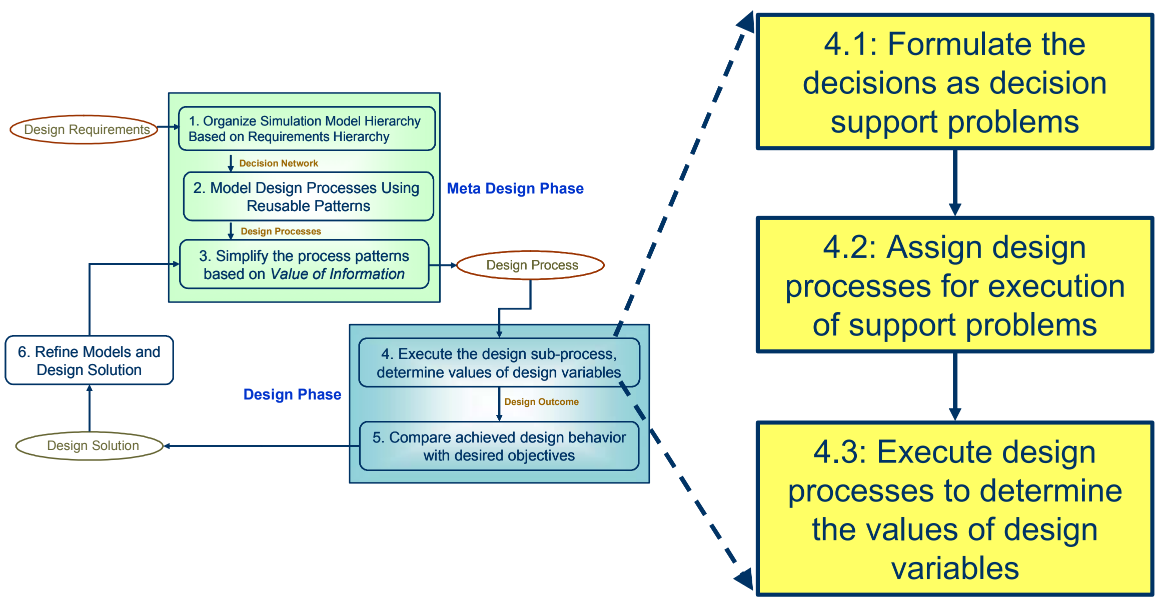


Figure 3-21 – Step 4 in the design method

Step 4.1: *Formulate the decisions as decision support problems:* The first step is to formulate the decisions using the decision problem constructs developed as a part of the DSP Technique. These constructs refer to the compromise and selection DSPs discussed in Section 2.2. The basic decision constructs are developed in (Bascaran, Bannerot et al. 1989; Reddy and Mistree 1992; Mistree, Hughes et al. 1993; Mistree, Lewis et al. 1994).

These basic constructs are extended for robust design (Chen 1995; Chen, Allen et al. 1996; Chen, Allen et al. 1997; Chen and Lewis 1999), fuzzy DSP (Allen 1996), interval based DSP (Reddy and Mistree 1992), utility based compromise and selection DSPs (Seepersad 2001), coupled DSPs, hierarchical DSPs (Koch 1997) etc. These extensions of decision support problems are used in different design scenarios where either different types of information are available, or different design considerations are important. For example, robust design is the case where the basic structure of the decision support problem remains the same but the manner in which the goals are formulated is different. In robust design, each goal in the compromise decision support problem is actually associated with two goals – achievement of target values and minimization of deviation from the target. These two goals can either be treated separately as two goals as carried out in (Chen, Allen et al. 1996) or can be combined together using a single design capability index (Simpson, Rosen et al. 1998). Similarly, in the case of utility based compromise decision, the manner in which the designers' preferences are formulated is different from the basic compromise DSP. Instead of the simple target values to goals, the preferences are mathematically formulated as utility functions (see Section 2.4).

In the case of patterns P1, P2, and P3 discussed in Section 3.5.2, there is a single decision supported by models at different scales. Hence, only a single compromise/selection DSP is required to model the decisions in these patterns. In the case of patterns P4, P5, and P6, there are two decisions (corresponding to the different scales) that may be coupled, independent, or may involve a sequential flow of information. In the case of patterns P7, P8, and P9, there are multiple decisions associated with each scale. Hence, at each scale, there are multiple decisions that may be coupled, independent, or sequential in nature. Each of these decisions is formulated using the selection or compromise DSPs. Although many different extensions of decisions can be used for formulating decisions in the different patterns, in this dissertation we focus our efforts on robust design based compromise decisions only. There are several categories of robust design, associated with different types of uncertainty (Type I through Type IV):

1. *Type I robust design*, originally proposed by Taguchi, centers on achieving insensitivity in performance with regard to noise factors—parameters that designers cannot control in a system. Relevant examples of noise factors are variation of ambient temperature, morphology changes, etc.
2. *Type II robust design*, proposed by Chen and coauthors (Chen, Allen et al. 1996), relates to insensitivity of a design to variability or uncertainty associated with design variables—parameters that a designer can control in a system.
3. *Type III robust design* (Choi, Austin et al. 2004; Choi 2005) considers sensitivity to uncertainty embedded within a model (i.e., model parameter/structure uncertainty). Model parameter/structure uncertainty is typically different from the uncertainty associated with noise and control factors, because it could exist in the parameters or structure of constraints, meta-models, engineering equations, and associated simulation or analysis models.
4. *Type IV robust design* (Choi, Austin et al. 2004; Choi 2005) is focused on uncertainty associated with design processes. Design process uncertainty emanates from the propagation and potential amplification of uncertainty due to the combined effect of analysis tasks performed in series or in parallel.

It is important to note that in all these four types of robust design, the main difference is in the design goals. Hence, the basic compromise DSP construct is adapted by changing the goals considered for design. In all the design problems used in this dissertation, we use robust compromise DSP for modeling product decisions. One of the primary reasons for adopting robust design approach is that the method is based on the idea of systematic refinement of both design process and the product. In other words, the designers start with a simple design process based on a number of assumptions and come up with a preliminary design. This preliminary design is then subsequently refined until the design requirements are met. Since the decisions are made in the presence of uncertainty due to simplified design processes and simple simulation models, the design decisions should be robust to uncertainties. After the decisions are formulated, the next step (Step 4.2) is to develop a design sub-process for executing that decision.

Step 4.2: *Assign design processes for execution of decision support problems (DSPs):* The decisions in the interaction patterns modeled in Step 4.1 are associated with design processes that represent solution schemes for corresponding decision problems. These design processes may either be composed of other decisions or may consist of elementary tasks that do not require design decisions. For single decisions in patterns P1, P2, and P3, the decisions can be executed by different types of algorithms such as exhaustive search, gradient based methods, genetic algorithms, etc. Each of these is associated with an elementary process that is followed and is embedded in the algorithm for executing the decision. Design decisions in simulation-based design are generally made using computer-based simulation models. A common augmentation of this process involves replacing the complex simulation models with simpler response surfaces to avoid large execution time. This adds a series of steps such as design of experiments, execution of complex model at various points in the design space, fitting a response surface, etc. to the process. These are simple elementary processes for making decisions.

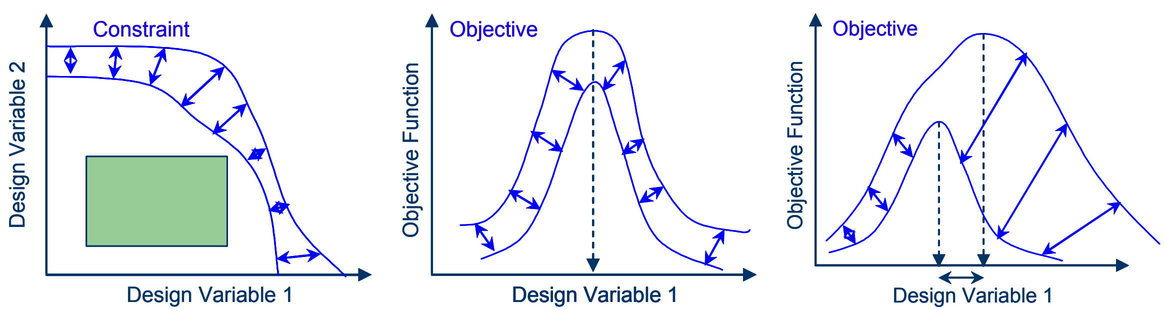
These processes become more complicated when multiple decisions are considered simultaneously. For example, in the case of sequential decisions in pattern P5, the process is a composition of two processes for execution of decisions in a serial manner. Similarly, in the case of pattern P6, two decisions are coupled with each other. These coupled decisions can be executed in following different ways – *a)* merging the two decisions into a single decision and solving the combined decision, *b)* passing ranges of solution from one decision to another, *c)* generating response surfaces of decisions that would be made by one designer as a function of decisions made by other decision (this response surface is also called rational reaction set) and then finding the intersection of these response surfaces, and *d)* making decisions iteratively until the final solution converges to a single point. Different processes have different characteristics and advantages, and therefore are suitable for different kinds of design scenarios. For example, the iterative process (option - d) may or may not converge to a point. The combination of multiple decisions into a single decision (option - a) results in the best design point, but is computationally expensive. The third option (option - c) is computationally less expensive but my result in an inferior solution. Hence, in this step of the design method, the designer assigns an appropriate process to the decisions formulated in Step 4.1. The design processes determined in this step (4.2) are then executed in Step 4.3.

Step 4.3: *Execute design processes to determine the values of design variables:* After the design processes are assigned to different decisions, the design processes are executed to determine the values of design variables. In order to perform the steps of design method in a computer interpretable manner, a design information modeling approach (3-P approach) is presented in this dissertation. The details of 3-P modeling approach are not presented in this chapter to maintain the logical flow. The approach is discussed in detail in Chapter 7 and the implementation details are discussed in Chapter 8. The approach is decision-problem based, and the main feature is the separation of declaration of decision problem related information (Step 4.1) and the information about processes to execute those decision problems. This supports reusability of design processes for different decision problems. In this chapter, we terminate the discussion of information modeling approach at this point and refer the readers to later chapters where it is discussed in detail.

The outcome of this step is a design for the product. This design consists of values for all design variables along with the performance achieved with the product. Due to the uncertainties inherent in the design process, the final design also has some uncertainty associated with it. If the design satisfies the requirements for expected behavior, then the designers can end the design process at this stage. However, if the requirements are not met and the uncertainty bounds are large (indicating that there is a possibility of refining the solution), then the design can be refined further through refinement of the simulation models. This is carried out in the sixth step in the design method, which is discussed in Section 3.5.5.

3.5.5 Targeted Model Refinement

All simulation models have some level of uncertainty associated with them because of the approximations and assumptions chosen during model development. However, this uncertainty may or may not affect the design decision. For example, in Figure 3-22, we present three illustrations in the impact of uncertainty on design decision. In the first case (Uncertain Constraint) the simulation model predicts the value of constraint; and due to the uncertainty inherent in the model, the constraint is represented as at range. The design space under consideration is shown as a rectangular area. Even with uncertainty, the constraint lies completely outside the design space. Hence, refinement of the model for reducing uncertainty would not affect the design decision. Similarly, the second case (Uncertain Objective - I) represents a scenario where the objective is to optimize an output of the simulation model. In this case also, the design decision (the value of design variable chosen to maximize the objective) is not affected. However, in the third case (Uncertain Objective - II), the uncertainty in the objective function results in different design solutions, making it important to refine the model further. The key notion here is to perform targeted refinement of simulation model, thereby efficiently utilizing the model development efforts.



a) Uncertain Constraint *b) Uncertain Objective - I* *c) Uncertain Objective - II*

Figure 3-22 - Need for targeted model refinement

Although the simulation models can be refined in a variety of ways such as consideration of additional physical phenomena, modeling some phenomena with greater accuracy, inclusion of interactions between different phenomena, refinement of mesh, making convergence criteria more stringent, use of better microscale model, and so on. All these factors have an effect on the accuracy of simulation models. It is obvious that the accuracy of simulation models is important for better design decisions. However, it is not the only factor to consider in decision making. The impact of inaccuracy in simulation models is not directly proportional to the impact on decisions. This is because of the impact of designers' preferences that act on the outputs of these simulation models. In other words, although there is inaccuracy in the prediction of a response, the designer's overall preference may not be sensitive to that response. The inaccuracy in simulation models may either get amplified or diminished by the designers' preferences. If the inaccuracy is amplified, then a small amount of inaccuracy has major impact on the overall result of the decision, and hence, the simulation model must be refined. In the case where inaccuracy is diminished, the inaccuracy in simulation models does not have strong impact on the overall result. Hence, the simulation model need not be refined.

There is an extremely broad spectrum of opportunities to develop methods for systematic refinement of models. In order to limit the scope of refinement methods in this dissertation, we focus on a small subset of model refinement. We assume that a simulation model has some parameters, whose values are uncertain. Information about upper and lower bounds on these parameters is available. Due to the uncertainty, these parameters can take any value between the lower and upper bound. Hence, in the context of this dissertation, we are only focusing on refinement of simulation models that corresponds to reduction in the range of possible values that these input parameters can take. Under this assumption that defines the scope of refinement discussed in this dissertation, we develop information economics based metrics (value of information) that quantify the impact of refinement of simulation models on the decision making. An overview of information economics is presented in Section 2.5. The value of information metric developed in this dissertation is discussed in detail in Chapter 4. It is important to note that refinement of simulation models is also a part of the design of design processes.

3.6 Role of Chapter 3 in This Dissertation

In this chapter, a method for the integrated design of products and design processes (RMS-DEM) is presented. The method is presented as an answer to the first research question posed in this dissertation. The method forms a basis for all the following chapters in the dissertation.

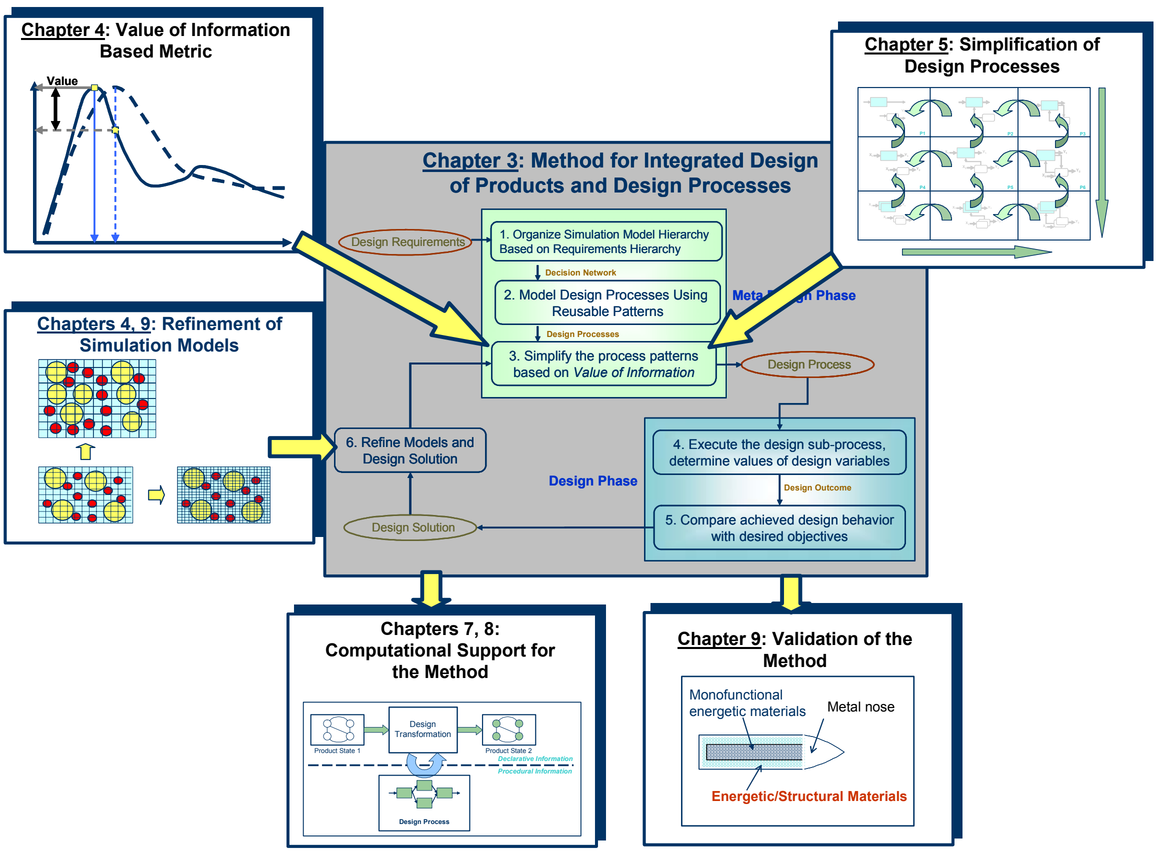


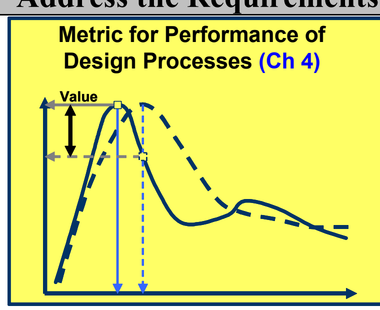
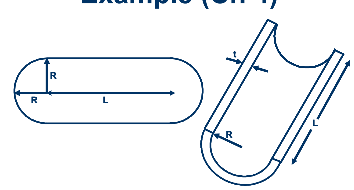
Figure 3-23 - Relationship of Chapter 3 with other chapters in the dissertation

The relationship of this chapter with the rest of the dissertation is presented in Figure 3-23. The details of Steps 3 and 6 in the method are discussed in Chapter 4 and Chapter 5, where the second research question is addressed. The method imposes certain requirements on the computational framework on which the design method can be implemented. These requirements are discussed in detail in Chapter 7 and an implementation strategy is presented in Chapter 8. The method is validated using a multiscale material design scenario in Chapter 9.

Chapter 4 Value of Information – A Metric for Making Decisions about Design Processes

In this chapter, we address the third requirement for the framework for integrated design of products and design processes: *"a metric for evaluating the performance of different design process alternatives"*. The six requirements for the framework are listed in Table 1-3. The components of the framework developed in this dissertation to address these requirements are highlighted in Table 1-6. A portion of Table 1-6 that is relevant to this chapter is reproduced on this page as Table 4-1. The component of the framework developed in this dissertation is a value of information based metric for assessing the performance of design processes. A pressure vessel example is used in this chapter to validate the metric. The metric is used for answering the second research question (RQ2) posed in this dissertation. The relationship of the value of information metric with RQ2 and the supporting hypotheses is presented in Section 4.1. Overviews of the role of value of information in designing design processes and the contents of this chapter are also provided in Section 4.1.

Table 4-1 – The requirement and component of the framework for integrated design of products and design processes addressed in Chapter 4

Framework Requirements	Components of the Framework Developed to Address the Requirements	Validation Examples
3) A metric for evaluating the performance of different design process alternatives		**Pressure Vessel Design Example (Ch 4)**  ***Purpose:*** To validate the value-of-information based metrics

4.1 Frame of Reference – Answering the Research Question 2 (Value of Information for Designing Design Processes)

Value of information refers to the impact of additional information on designers' decision making capability. Different design process options can be compared based on the quality of product decisions that designers can make in terms. This difference in quality of decisions made by different process options is quantified by the value of information metric. As discussed in Section 1.2.2, value of information is only one of the metrics based on which design processes can be analyzed. Other metrics include execution time, cost, complexity, modularity, robustness, etc. Since the focus of this dissertation is to design processes from a decision centric perspective, we believe that value of information is one of the most important metrics for comparing different process options. Other metrics can be developed and included in the design method described in Section 3.5.

In this chapter, the goal is to develop and validate a value of information metric for making design process decisions. This metric is fundamental to answering the second research question which is: "How should multiscale design processes be systematically simplified and models refined in a targeted manner to support quick design decision making without compromising the decision quality?" The first hypothesis (H2.1) to support the answer to this question is that "design processes can be simplified and models refined by making tradeoffs between the value of information obtained via simulations and need to achieve robust, satisficing solutions". Simplification of design processes eliminates some information that could be used for decision making. Similarly, refinement of simulation models adds information that improves the designers' decision making capability. The metric developed in this chapter are useful for quantifying this improvement in decision making capability. The impact of refinement of simulation models is discussed in this chapter. The value of information metrics developed in this chapter are used in Steps 3 and 6 of the design method proposed in Chapter 3 (see Figure 4-1). The research questions and hypotheses addressed in this chapter are highlighted in Figure 4-2. The validation example used in this chapter is that of a pressure vessel design problem. The concepts presented in this chapter form a basis for the simplification of design processes presented in Chapter 5.

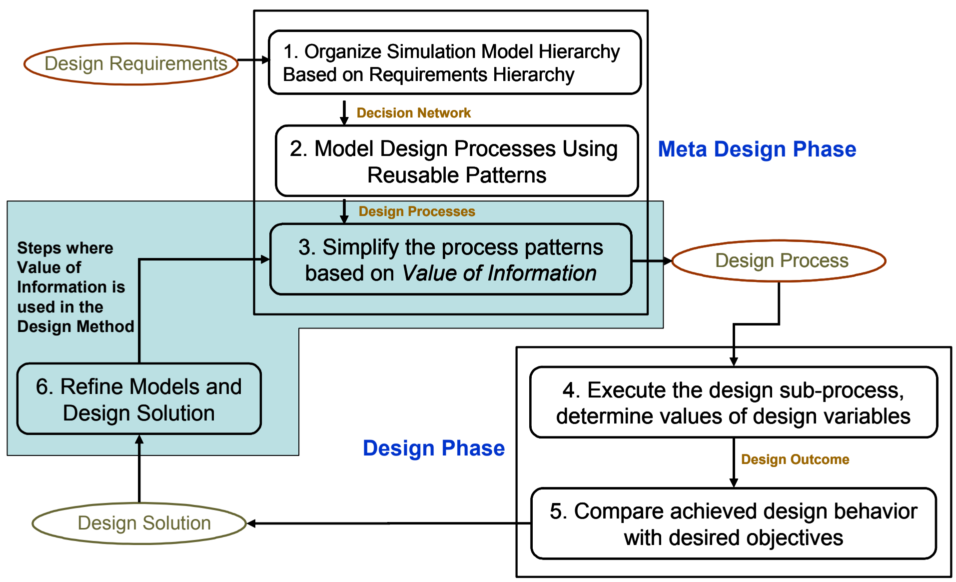


Figure 4-1 – Role of value of information metric in the design method

Metrics for value of information is used in previous research efforts in information economics. A review of the existing literature in quantifying value of information is presented in Section 4.2. A critical evaluation of this literature is centered on designers' requirements for making meta-level decisions. From the literature review, it is observed that the existing metrics for value of information do not account for all the aspects of decisions made in meta-design. Existing metrics are only based on the additional information that changes knowledge about the probability of occurence of random events.

Further, existing metrics compare scenarios with and without additional information only based on the expected payoff of the outcome. These limitations are discussed in detail in Section 4.2 along with the requirements for value of information metric to be used in meta-design.

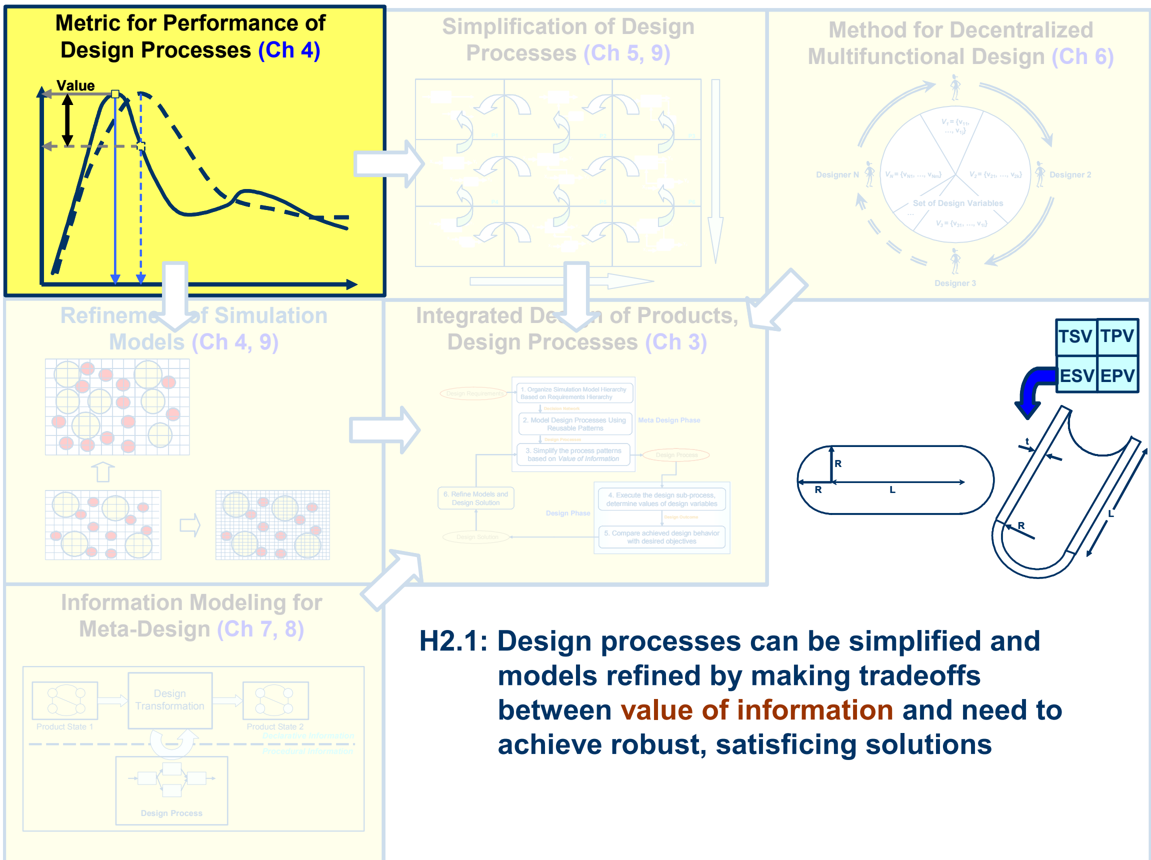


Figure 4-2 – Research hypotheses addressed in Chapter 4

A new metric for value of information is presented in Section 4.3. The metric is consists of three components – ex-post range, opportunity ratio, and achievement ratio. Each of these three components of the metric quantifies different aspects of the impact of additional information. The details of these components are discussed with the implications on design process related decisions. The metrics are applied to a problem of designing pressure vessels in the presence of imprecision about material properties in

Section 4.4. The advantages and limitations of using the value of information developed in this dissertation are discussed in Section 4.5.

4.2 Value of Information for Decision Making – A Literature Review

At any stage in the decision making process, designers possess some amount of information that can be used for selecting the best course of action. Designers have an option of either *i)* making a decision using the available information, or *ii)* gathering more information and then making a decision using the updated information. In this context of decision making, value of this added information refers to the *improvement in designers' decision making capability*. The value of information metric is used by decision makers to make the meta-level decision involving tradeoff between gathering more information to reduce uncertainty and reducing the associated cost. The idea of using value of information for determining whether to consider additional information for decision making is not new. It was first introduced by Howard (Howard 1966). The expected value of information as defined by Howard is *"the difference between the expected value of the objective for the option selected with the benefit of the information less than without"*. Mathematically, the expected value of information is given by:

$$EVI_{\varphi} = \underset{\gamma}{E}\left[\max_{i}\left\{\underset{p}{E}[u(p, x^{(i)}) \mid \varphi]\right\}\right] - \max_{i}\left\{\underset{p}{E}[u(p, x^{(i)})]\right\}$$

where, EVI_{φ} is the expected value of information, φ is the available information, $\underset{x}{E}[f(x) \mid \varphi]$ is the expected value of the function $f(x)$.

Bradley and Agogino use this value of information metric for catalog selection problem, where a designer is faced with the task of choosing components from a catalog in order to satisfy some functional requirements (Bradley and Agogino 1994). During the conceptual design phase, selection decisions are characterized by significant uncertainty due to limited understanding of requirements and constraints, inability to specify part dimensions, uncertainty in the environmental conditions, etc. However, before making the decision about the right component, a designer need to make *another higher level decision* – whether to go ahead and make the decision using available information or to spend resources and gather more information before making the selection decision. This is a meta-level decision, for which Bradley and Agogino (Bradley and Agogino 1994) utilize the value of information metric to quantify the expected benefit from additional information.

Poh and Horvitz use the value of information metric for refining decisions (Poh and Horvitz 1993). The authors present three dimensions in which the decision models can be refined – quantitative, conceptual, and structural. Quantitative refinement of a decision model can be carried out by reducing the uncertainty in the decisions problem or by refining the preference models. Conceptual refinement is carried out by refining the definition of alternatives and design variables, whereas structural refinement requires addition of dependencies in the simulation model. Poh and Horvitz use the value of information metric to determine which dimension is critical for refinement of the decision problem.

Lawrence provides a comprehensive overview of metrics for value of information (Lawrence 1999). He argues that the value of information for decision making can be measured at different stages in the decision-making process. Accordingly, the value of information metrics are named differently based on the stage at which it is evaluated. Four different options available for measuring value of information are:

a) prior to consideration of incorporation of information,

b) Ex-ante value: after considering a message source but prior to receiving a message,

c) Conditional value: after receiving additional information and making the decision, but before realization of the environmental state, or

d) Ex-post value: after addition of information and making a decision-based on acquired information.

Determination of value of information at different stages in the decision-making process results in different kinds of insight for meta-level decisions. The appropriateness of a stage for measuing value of information depends on the problem at hand and the available information. Consider an example of a designer who has a simulation model for predicting the system behavior and is interested in making a decision using the model. Before making the decision, he/she has an option of increasing the fidelity of the model by considering additional physical phenomena in the model. For example, a structural analyst may improve the fidelity of static model by adding dynamic behavior, creep, etc. Description of a physical phenomenon is equivalent to an information source that generates information about the system behavior. The output of the simulation i.e., system behavior is equivalent to the added information generated by the information source. Now, the decision maker can evaluate the expected value of information before even considering the incorporation of additional any physical phenomena. The second option (*ex-ante* value) is to decide which physical phenomena to model a specific physical phenomenon (i.e., information source) and evaluating the value of information before executing the simulation code. The third option (*conditional* value) is to evaluate the value after executing the simulation code and making decisions about the system but before manufacturing and testing system. In this scenario, there is uncertainty in the actual system behavior that would be achieved due to factors such as manufacturing variability and changes in environmental conditions. The fourth option (*ex-post* value) is to evaluate the value of this additional information after making decisions and also manufacturing and testing the system. In this scenario, the designers know exactly how the system behaves.

Mathematically, the ex-post and ex-ante value of information are represented as follows:

1. *Ex-post value*: $v(x, y) = \pi(x, a_y) - \pi(x, a_0)$,

   Where, a_0 and a_y represent the actions taken by decision maker in the absence and presence of information y. $\pi(x, a)$ represents the payoff achieved by selecting an action a, when the state realized by the environment is x.

2. *Ex-ante value*: $v(x, y) = E_{x|y}\pi(x, a_y) - E_x\pi(x, a_0)$, where $E_x f(x)$ is the expected value of $f(x)$ and $E_{x|y} f(x)$ is the expected value of $f(x)$ given y. It is important to realize that the key difference between ex-post and ex-ante value is that in ex-post value, the realization of state x is known. However, the realization of state x is not known in ex-ante value and the expected value of payoff is taken over the uncertain range of state x.

Ideally, the designers are interested in the *ex-post* value of information because it truly reflects the value of information for a decision-based on the actual behavior of the system. There system behavior is known deterministically. However, it is not possible to calculate the ex-post value of decision before making the decision itself. Due to the ex-ante nature of decision making, the decisions about the information have to be made before the state actually occurs. Hence, the *ex-ante* value of information is generally used by designers. It captures the value of information by considering uncertainties in the system.

In order to model uncertainty for evaluating value of information, it is assumed that the probability distributions are available. However, if these probability distributions are not available, they are generally generated through an *educated guess* that is based on the designers' prior knowledge. In order to address the problem of lack of knowledge about the probability distributions, Aughenbaugh and co-authors (Aughenbaugh, Ling et al. 2005) present an approach of measuring the value of information based on probability bounds. They assume that although the exact probability distributions are unavailable, the lower and upper bounds on these probability distributions are available in terms of p-boxes. Using this p-box approach, they evaluate the value of added information that reduces the size of the interval for probability distribution (i.e., tightens the bounds on the p-box).

Critical Review and Requirements for the Value of Information Metric

In most of the efforts, the value of information is based on the variability in the decision problem. This uncertainty is modeled using probability distributions. However, except for Aughenbaugh and co-authors (Aughenbaugh, Ling et al. 2005), imprecision in the decision models which cannot be modeled in terms of probability distribution functions is generally not modeled. Imprecision relates to epistemic uncertainty (i.e., the lack of knowledge), whereas variability refers to aleatory uncertainty (i.e., inherent randomness in the system). The key difference between imprecision and uncertainty from a value of information standpoint is that imprecision can be reduced by incorporation of more information but uncertainty cannot be reduced via incorporation of information. For example, consider a scenario where a designer has an option of making a decision using one of the two available simulation models. One of the simulation models has a higher fidelity representation of physics than the other. The meta-level decision that the designer has to make is – "*Which simulation model should he/she use for making the decision?*" This scenario is extremely common in multi-scale design problems. Consideration of imprecision in the value of information in addition to uncertainty is important from a meta-design standpoint and is used for developing methods for systematic simplification in the next chapter. It forms a basis for determining the extent of refinement of simulation models in later chapters. Hence, the first requirement for value of information metric is *quantification of imprecision in addition to statistical variability*.

Further, in most of the efforts at utilizing the value of information, only the increase in expected value of payoff due to added information is considered for making meta-level decisions. The variation in utility due to uncertainty is not considered. For example, consider the scenario shown in Figure 4-3. In this figure, a designer needs to select between two alternatives A and B. Parts (a) and (b) of the figure represent the distributions of payoff values for the two alternatives before and after the addition of information respectively. The variation in utility shows the impact of uncertainty and imprecision on achievable payoff values. As shown in the figure, the expected value of payoff achieved by each alternative is the same before and after addition of the information. However, after addition of information, the range of utility values that can be achieved by each alternative has reduced, which implies that the decision maker has a greater confidence that Alternative A performs better than Alternative B. Hence, designer's decision making capability is increased by the addition of information. If the decision about addition of information is made only based on the increase in expected value of utility, then it does not reflect the increased decision making capability of the designer. An increase in confidence of a decision maker increases the decision making capability. Hence, *the reduction in the deviation in expected value should also be considered in value of information metric*. This is the second requirement for the value of information metric developed in this dissertation.

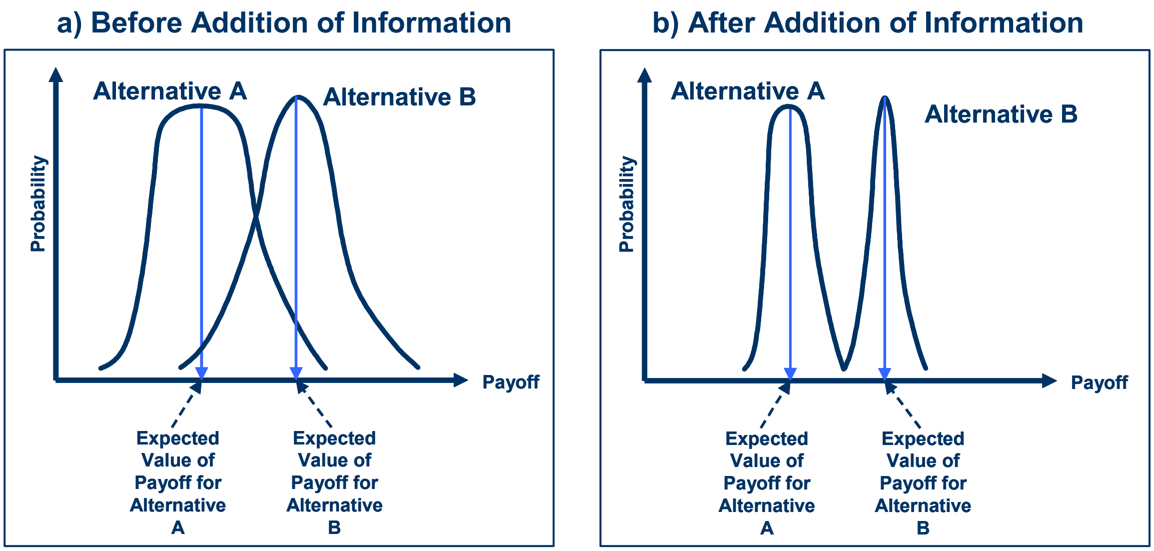


Figure 4-3 – Importance of including deviation of expected utility in calculating value of information

The third requirement for the value of information metric is to *quantify the opportunity for improving the design solution by adding more information*. That is, the value of information metric should quantify the upper bound on benefit that can be achieved by obtaining perfect information. The opportunity for improving the design solution is quantified in the literature using Expected Value of Perfect Information (EVPI), which is calculated as the expected value of information based on the setting exact values of uncertain parameters. In there are n uncertain parameters, the expected value of perfect information corresponding to each parameter is evaluated for individual parameters by setting their exact values. The greater the expected value of perfect information, greater is the opportunity of improving the design solution through information gathering. The limitation of this expected value of perfect information, however is that the exact values of parameters is generally not available before gathering the information. Hence, the requirement is that the metric should provide an indication of the opportunity without requiring the perfect information.

The requirements for quantifying the value of information in improving decision making capability are summarized in Table. In Section 4.3, we present a value of information metric that satisfies these requirements.

Table 4-2 – Requirements for the value of information metric

1. Quantification of imprecision as well as statistical variability in available information
2. Consideration of the deviation in payoff function in addition to the expected value
3. Quantification of opportunity for improving the design solution by adding more information

4.3 Proposed Metric for Value of Information for Decision Making

Since our focus in this dissertation is on simulation-based design, a simulation model is the source of information in the context of this dissertation. Simulation models are used to predict the behavior of systems, which in turn are used for decision making.

Simulation models inherently contain some inaccuracy due to the assumptions in the model. Addition of more information for decision making is equivalent to refining the simulation models so that they are closer to the exact behavior of the system. By developing the value of information metric, we are interested in providing support to the designers by quantifying the need to refine simulation models in a systematic manner.

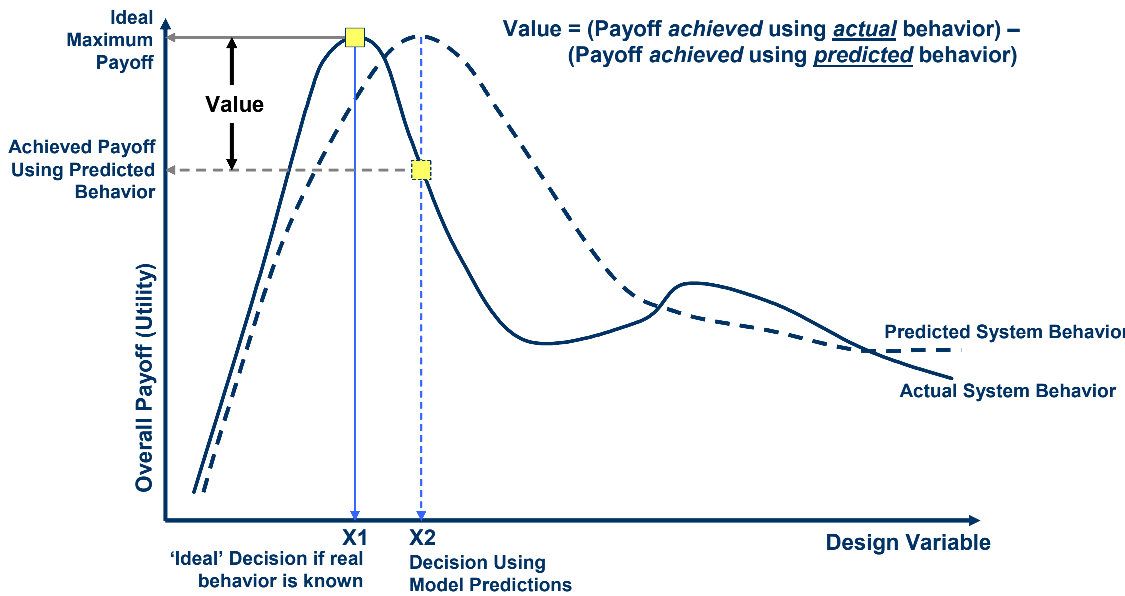


Figure 4-4 Conceptual description of value of information in simplified models

Consider a scenario shown in Figure 4-4, where the horizontal axis is the value of design variable and the vertical axis is the corresponding payoff that is achieved by selecting the design variable. The design variable can be some physical dimensions that the designer has control over, whereas the payoff represents profit, which depends on system behavior such as performance, strength, and cost. The designer's objective is to maximize the payoff by appropriate selection of the design variable value. The solid line represents the *actual* system behavior and the dashed line represents the system behavior *predicted* by the simulation model. The difference in actual and predicted behavior is due to the imprecision in the model. In Figure 4-4, it is assumed that the decision is characterized by no statistical variability, but only imprecision due to the inaccuracy in the simulation model.

If a designer makes a decision only using the simulation model, the decision point is X2, because it maximizes the payoff based on the predicted behavior. However a designer would have selected decision point - X1 if the actual (real) behavior of the system were known (by using a *perfect* model). Hence, the value of using the perfect model over simpler model is the difference in payoff *actually achieved* by using the exact model and the payoff achieved by making decisions using simpler model. It is important to note that the value of information is evaluated using the difference in payoff using the actual system behavior. This value of information is in close agreement with the ex-post value used in the literature.

In order to illustrate the point, three hypothetical scenarios with different predicted behavior resulting from different simulation models are shown in Figure 4-5, Figure 4-6, and Figure 4-7. In Scenario 1, the decision of design variable made by using the simpler model and the exact model is the same which implies that the difference between payoff achieved using the simple model and exact model is zero. Hence, the value of using the exact model over simpler model is zero. In Scenarios 2 and 3, the value of design variable as selected by the simpler model is different from the decision using exact models. Hence, the value in both scenarios is non-zero. However, the magnitude of value is different in both Scenarios. When compared with Scenario 2, the value is greater in Scenario 3.

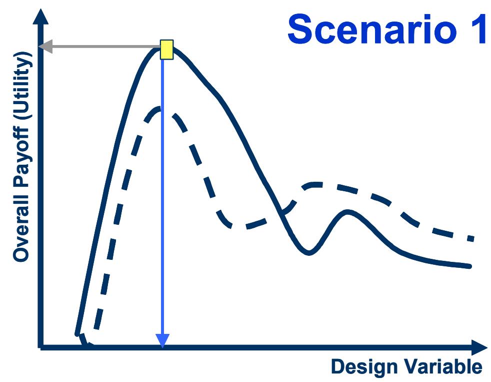


Figure 4-5 – Illustration of value of information – scenario 1 (decision made using predicted system behavior is the same as decision made using actual system behavior, value = 0)

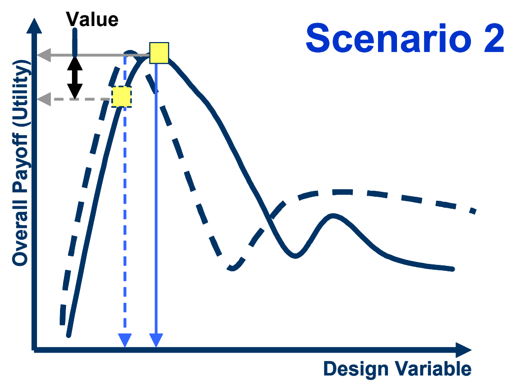


Figure 4-6 - Illustration of value of information - scenario 2 (decision made using predicted system behavior is close to the decision made using actual system behavior)

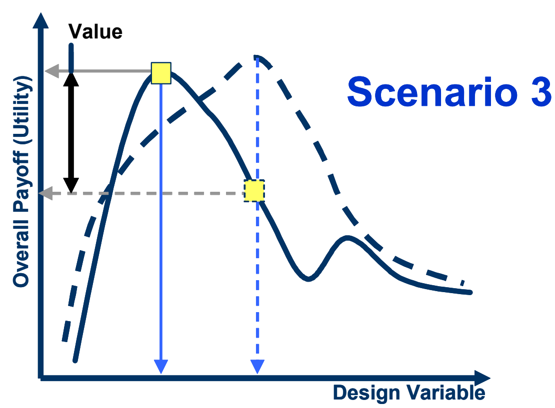


Figure 4-7 - Illustration of value of information - scenario 3 (decision made using predicted system behavior is far from the decision made using actual system behavior)

Another important point that the value of information metric highlights is that the value does not depend only on the accuracy of the model. It also depends on the complete decision formulation that includes constraints, preferences, region in the design space that is under consideration, etc. This point will be illustrated further in Section 4.4 using a design example. The same concept extends to higher dimension problems where there are many design variables and the payoff is determined by multiple conflicting criteria. In the case of multiple design variables, the curve corresponds to a multidimensional surface. In the case of multiple design criteria that affect the payoff, the criteria are combined together into an overall payoff function based on designers' preferences.

Payoff Functions using Utility Theory

In the discussion so far in this chapter, we have based the discussion of evaluating value of information on difference in payoff functions. The question that arises is how to evaluate this payoff function. In the field of economics, the payoff is determined in terms of the overall monetary profit. However, in engineering design, designers are generally faced with multiple criteria that affect the decision about design variables. It may not be possible to express all criteria in terms of monetary benefits. Hence, these multiple criteria are converted into value functions that represent a decision maker's preference for different levels of criteria on which the decision is based. These value functions associate a real number $v(x)$ to each value of the criteria x (Keeney and Raiffa 1976). Two different values of the criterion x have equal value v if they are equally preferable. Since the value function is used to quantify designer's payoff, a decision problem entails maximization of the value function. Although there are many different ways in which value functions are determined, a special type of value function called *utility function* is commonly used in engineering design decisions because of its capability to handle *a)* uncertainty in a mathematically consistent manner and *b)* multiple decision criteria. Utility functions and the axioms on which they are based are proposed by Neumann and Morgenstern (Von Neumann and Morgenstern 1947). The details of utility functions are discussed in details elsewhere in Keeney and Raiffa (Keeney and Raiffa 1976), Hazelrigg (Hazelrigg 1998). In this dissertation, we assume that the payoff values for multiple decision criteria in the presence of uncertainty are quantified in terms of utility functions and the decision is made by maximizing the expected value of utility.

As mentioned in the previous discussion, Figure 4-4 represents a scenario characterized by imprecision only. If there is uncertainty associated with the problem in addition to the imprecision, then corresponding to each design variable value, there is a probability distribution of payoffs that can be achieved. The decision in such a case will be based on maximization of the expected value of payoff. This is illustrated in Figure 4-8 where both the actual system behavior and the predicted behavior are associated with uncertainties represented by probability distribution functions. The solid and dashed lines for system behavior in Figure 4-8 correspond to the expected values of payoff. The key difference in decision making scenarios with and without uncertainty is that instead of the deterministic value of payoff, a probability function of payoff is known. Hence, instead of maximizing a discrete value, the expected value of payoff is maximized. While keeping this key difference into consideration, we go ahead and develop the value of information metric for a case with only imprecision and we later extend that to scenarios where uncertainty is present.

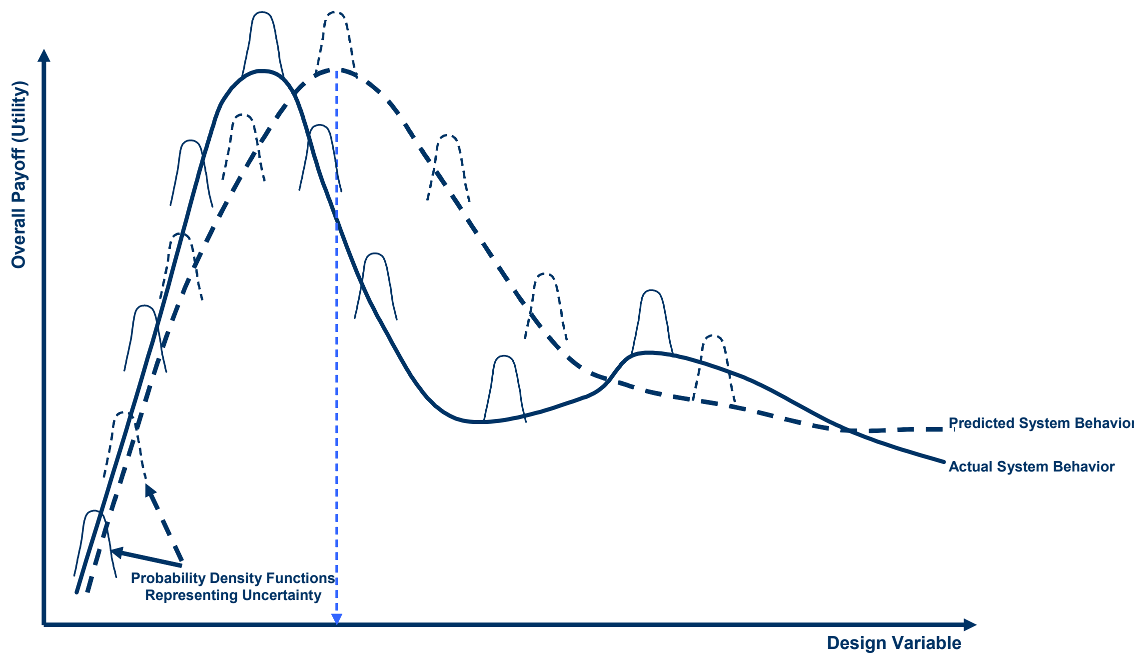


Figure 4-8 Decisions characterized with imprecision and variability

Given that the utility functions will be used for measuring the payoff, the value of information is defined as the increase in the achievement of overall utility value when an exact model is used as compared to simplified model. However, in most design cases, the difficulty is that the exact system behavior as shown conceptually in Figure 4-4 is seldom available. If the exact system behavior is available, there is no need to use the simulation model to predict the behavior. (Note that in some cases, even if the exact system behavior is available, it may not be used for decision making because of the associated costs.) In many design cases, although the real system behavior is not available, it may be possible to determine the upper and lower bound on the behavior predicted by simulation model. The designers may be able to generate information about lower and upper bounds through physical experiments, or though analysts' insight into the system's behavior. These bounds on the imprecision of the model result in bounds on the overall utility function, as shown in Figure 4-9. The availability of bounds on imprecision of simulation models is an assumption, based on which the value of information metric is developed in this dissertation.

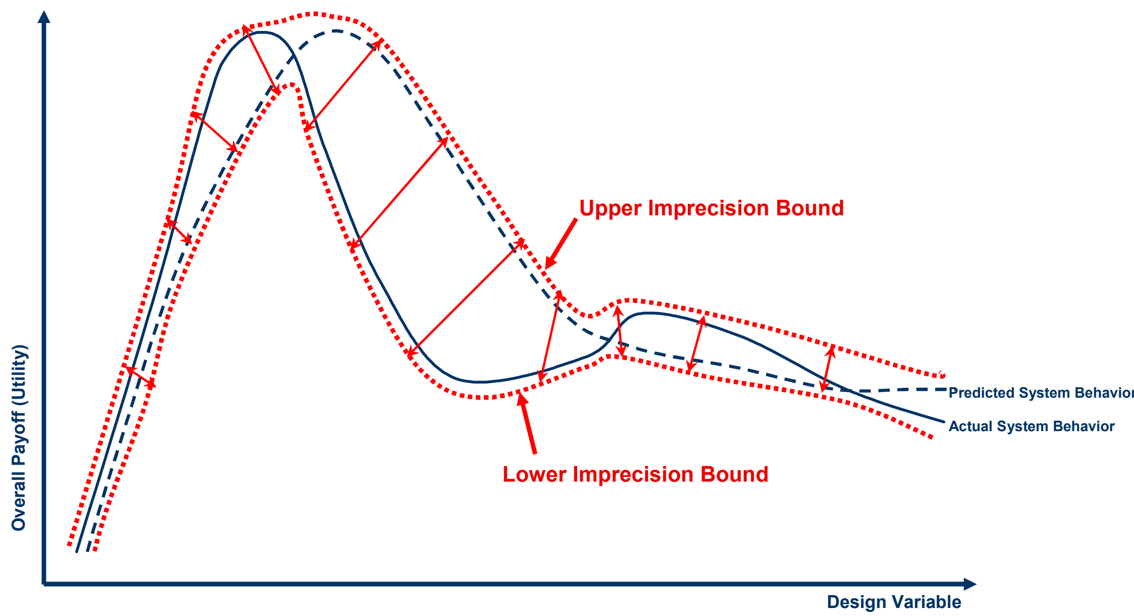


Figure 4-9 – Bounds on imprecision of utility function due to imprecision in simulation model

With the available information about lower and upper bounds on payoff function value, the decision maker can select a decision rule based on which he/she selects numerical values for the design variables. The decision rule can be *a)* maximize the lower bound on achievable payoff (i.e, the worst case scenario), *b)* maximize the upper bound on achievable payoff (i.e., best possible scenario), or *c)* maximize the average value of payoff. In Figure 4-10, the decision maker's decision rule is to maximize the average payoff, based on which the numerical value of design variable shown in the figure. For the selected value of design variable, there is a range of achievable payoffs as a result of imprecision in the simulation model. The lower bound of achievable payoff is denoted by 'l', the upper bound by 'u', and the average value by 'm'. The maximum payoff that can possibly be achieved by any value of the design space is denoted by 'p', and is evaluated by maximizing the upper imprecision bound on payoff. Since the exact value of the payoff is not known at different values of design variables, it is not possible to calculate the exact value-of-information as illustrated in Figure 4-4. However, since the lower and upper bounds on payoff are known throughout the design space, we can determine the *maximum possible value-of-information*. This **upper bound on the value-of-information** (maximum possible ex-post value) is given by the difference **(p-l)**, where 'p' is the maximum payoff that can be achieved by any point in the design space and 'l' is the lowest possible payoff value achieved by the selected point in the design space (after making the decision without added information). This upper bound is also referred to as the **ex-post value range**.

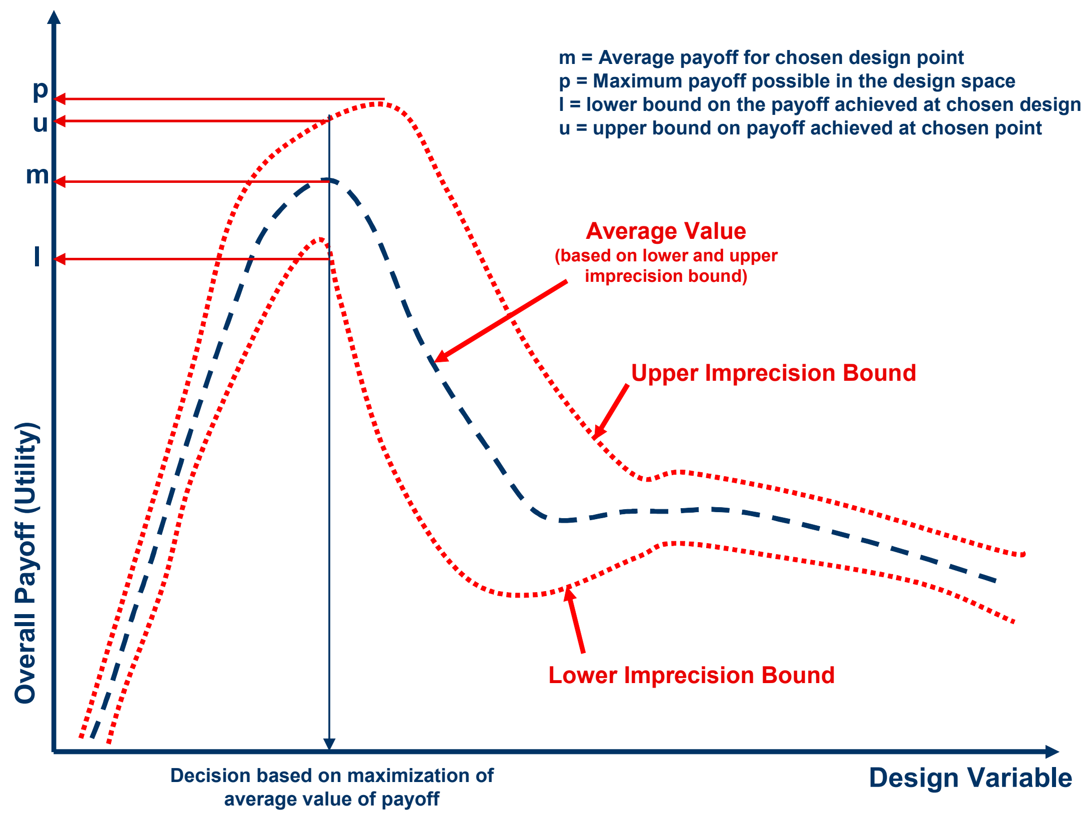


Figure 4-10 - Decision made using bounds on payoff

The decision on whether or not more information should be gathered is made based on the upper bound on value of information. It is also important to assess the impact of additional information. By addition of more information for decision through refining the simulation model, the range of achievable utilities as represented by imprecision bounds becomes smaller. Due to the reduction of bounds, the designers obtain better confidence of the utility that can be achieved. In some cases, the decision point may shift, possibly resulting in a better decision. These two effects of addition of information are captured in the following two ratios –

1. *Opportunity Ratio* (R1), which indicates how good the current design point is, when compared to the best possible utility value. It is evaluated using the following relation:

$$R1 = \left(\frac{u-m}{p-m} \right),$$

where, 'u' is the upper bound of payoff achieved, at the chosen design point, 'm' is the average payoff achieved at the chosen design point, 'p'is the maximum possible payoff throughout the design space. If $p = m, \text{then} R1 = 1$.

2. *Achievement Ratio* (R2), which indicates how good the current design point is, as compared to what the designer wants. In order to evaluate this ratio, a designer pre-defines a reference (cutoff) value for overall utility, on the achievement of which the designer is reasonably satisfied. This reference value of utility is denoted by 'r'. The achievement ratio is given mathematically as:

$$R2 = \left(\frac{u-r}{u-l} \right)$$

where, 'u' is the upper bound of payoff achieved, at the chosen design point, 'l'is the lower possible payoff at the selected design point. If $u = l, \text{then} R2 = 1$.

The opportunity and achievement ratios lie between 0 and 1. Low values of R1 indicate that there is a hope of getting a better solution by addition of information due to the possibility that design solution may move to better regions in the design space. However, high values of R1 indicate that there is low hope of improvement of design solution by moving into better points in the design space. Similarly, lower values of the achievement ratio (R2) indicate that the current decision point does not meet the satisfaction level of the designer in terms of reference value for overall utility. Higher values of R2 indicate achievement of designer's expectations. This is shown graphically in Figure 4-11, where the two ratios are plotted on horizontal and vertical axes. The graph is divided into four regions – A, B, C, and D. Region A represents the case where both R1 and R2 are low. Hence in this region, the designer has not achieved what he/she wants but there is an opportunity (hope) of achieving better solution. Hence, there is a need to gather more information. The value of information is high in this region. In region B, designer's wishes are not fulfilled and there is little opportunity of improving the solution. Hence, the value of information is low in region B. In region C, designer has achieved what he/she wanted as specified in the reference value but there is also hope of improvement of design solution. The value of information is high (but is lower that value of information in A). Similarly, in region D, the achieved utility is close to the reference value and there is little hope of achieving a better solution. Hence, the value of information is low. In other words, the two ratios combined together with the ex-post range serve as metrics for value of information.

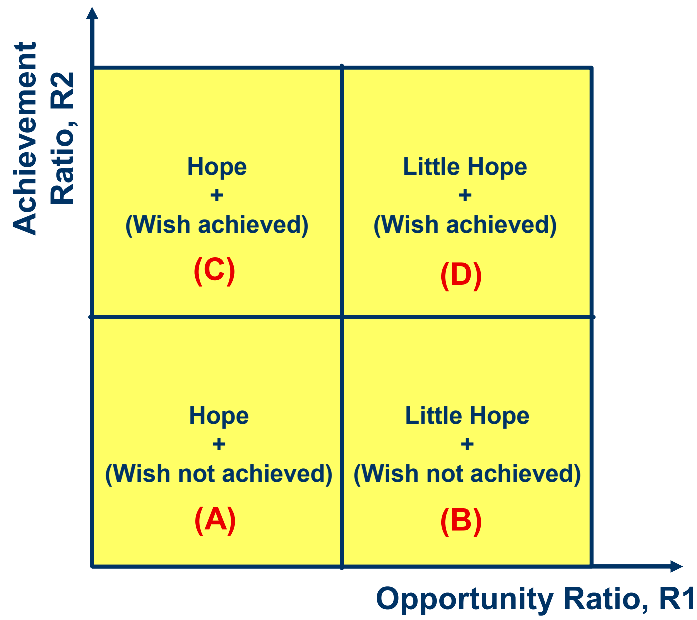


Figure 4-11 – Segmentation of opportunity and achievement ratios to denote opportunity and achievement

Each of these three metrics for value of information highlights different aspects of improvement in decision making capability by addition of more information. Ex-post range indicates the maximum possible increase in utility value if the information is added to the decision. Opportunity Ratio indicates the possibility of achieving a better solution in other regions of the design space. This improvement in solution is possible only if information is added. Achievement ratio indicates how well the currently selected design point satisfies designer's objectives. The implications of different values of each of these metrics and possible actions by designers are listed in Table 4-3. When making a meta-level decision to gather more information, the designers should consider all three metrics and use their judgment considering the decision problem at hand.

Note that although the metrics are developed for information generated by simulation models, it is applicable to other kinds of information such as that generated by physical models as long as the following assumption is valid - the predicted value and associated error in terms of lower and upper bounds are available. In the next section, we apply the value of information metrics proposed in terms of upper-bound on ex-post value, opportunity ratio and achievement ratio to a pressure vessel design problem and show the benefits from using these metrics.

Table 4-3 - Implications of combinations of different levels of value metrics and the corresponding actions suggested for designers

Ex-Post Value	Achievement Ratio	Opportunity Ratio	Implications	Designer's Action
High	*High*	*High*	Little hope + wish achieved + low confidence	Adding more information will improve the confidence but is not necessary
High	*High*	*Low*	Hope + wish achieved + low confidence	Addition of information may improve both solution and confidence but is not required
High	*Low*	*High*	Little hope + wish not achieved + low confidence	Adding more information may result in better solution
High	*Low*	*Low*	Hope + wish not achieved + low confidence	Addition of more information may take the solution to other point and also increase confidence
Low	*High*	*High*	Little hope + wish achieved + high confidence	No need to improve the model
Low	*High*	*Low*	Hope + wish achieved + high confidence	Addition of more information may result in a better solution
Low	*Low*	*High*	Little hope + wish not achieved + high confidence	Addition of more information will not help
Low	*Low*	*Low*	Hope + wish not achieved + high confidence	Addition of more information may result in new points. Designer should add more information because the current solution is not good

4.4 Example – Pressure Vessel Design

In this section, we discuss an example design problem where the objective is to design a pressure vessel with low weight and high volume. The pressure vessel should be able to sustain a pre-specified pressure. The density and yield strength of the material are determined using some (unspecified) material simulation model. The accuracy of the material simulation model can be improved by addition of more details about the microstructure properties. However, this addition of information requires costly experiments and it is desired to keep the simulation model as simple as possible. Due to the simplicity of the material model, the material properties are inaccurate. An accuracy bound on the values predicted by the simulation model is available. In other words, although the predicted value has some errors, it is known with confidence that the values will lie between associated lower and upper bounds. It is important to note here that the imprecision in the model is not due to randomness but is a result of lack of knowledge about the system. As the accuracy of the material simulation model is increased, the bounds on predicted values decrease and the predicted value of properties change.

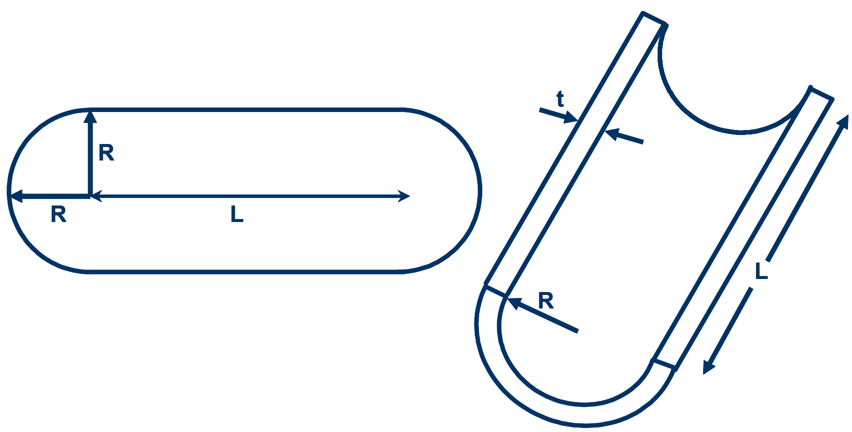


Figure 4-12 – Thin walled pressure vessel (Lewis and Mistree 1997)

Using the material information model, a designer intends to determine the following dimensions of the pressure vessel - radius, length and the thickness (see Figure 4-12). In this problem, the values of length and radius are fixed in order to make it a one dimensional problem that can be easily visualized using 2-dimensional plots. Hence, the design variable is the radius of pressure vessel. Due to the manufacturing constraints, there are limitations on maximum and minimum values of these dimensions. There are additional constraints on the design problem such as – a) the pressure should not fail under the given pressure, b) the pressure vessel is thin walled, thereby imposing geometric limitations. This problem is adapted from (Marston 2000). The compromise DSP formulation of the problem is provided in Table 4-4. The decision problem is graphically shown in Figure 4-13 as a utility maximization problem with inputs of preferences, constraints, goals and associated targets, and the design variable. The material properties are shown with block arrows depicting the imprecision due to simplified material models. Due to this imprecision in material properties, the overall utility is imprecise, and hence represented by a block arrow.

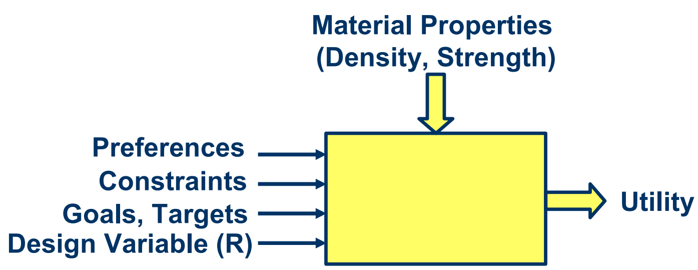


Figure 4-13 – Spring design problem with imprecise material properties leading to imprecise utility

A designer's preferences are modeled using risk averse utility functions. The mathematical relationships for utility functions are shown in the mathematical formulation of cDSP in Table 4-4. These utility functions for weight and volume goals are graphically shown in Figure 4-14 and Figure 4-15 respectively.

Since the decision about the radius of pressure vessel is made under imprecision about material properties, the decision is formulated not only to maximize the mean achievement value of the utility, but also to minimize the range of possible utility values achievable using the selected decision point. Hence, in addition to the utility functions for volume and weight, two utility functions for average utility value and the difference between maximum and minimum possible utility values for the chosen point are formulated. The overall objective function is formulated as a weighted sum of these two utility functions. The decision criterion used in this formulation is to select the point in design space that maximizes this objective function. The objective function can also be formulated in other forms such as the minimum possible utility, weighted sum of minimum and maximum possible values. Maximizing the minimum possible utility selects the point in design space where the worst case performance is the best, and leads to a conservative design. We believe that the selection of the decision criterion (i.e., the objective function) is dependent on the problem at hand and should be chosen by the designer. Although the results will be different for different decision criteria, the general principles and the discussion that follows remain valid and are independent of the decision criterion chosen.

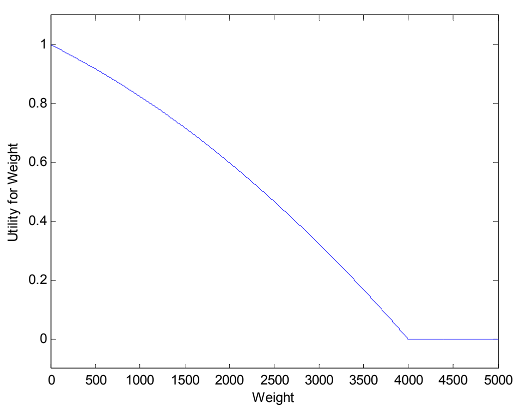


Figure 4-14 – Utility for weight

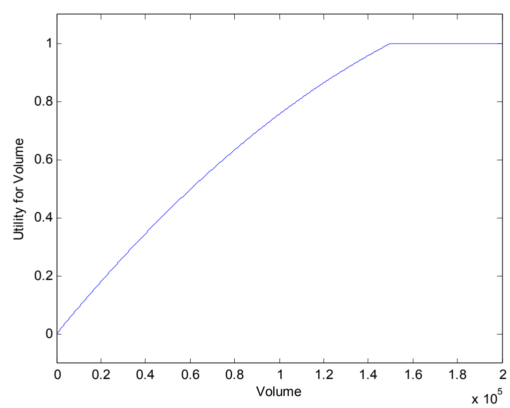


Figure 4-15 – Utility for volume

The decision formulation discussed so far in this section is the design level decision about the product. However, there is also a meta-design level decision that needs to be made in this problem – determination of the maximum level of imprecision in the material model which is appropriate for making decisions about the pressure vessel.

Beyond that maximum level of imprecision, the cost of reducing imprecision overtakes its potential value. We address the problem in two parts – one where only the density is imprecise and the other where the material's strength is imprecise. These two parts are discussed in Sections 4.4.1 and 4.4.2 respectively. This meta-level decision is made using the value of information metric defined in Section 4.3.

Table 4-4 - Utility-based Compromise DSP of pressure vessel design decision

Utility-based Compromise DSP of pressure vessel design decision
Given *Strength (S_t), Pressure (P), Density(ρ)* Length (L) Thickness (T) Imprecision in the material model *Some helpful relations:* Volume, V = $\pi\left[\frac{4}{3}R^3 + R^2L\right]$ Weight, W = $\pi\rho\left[\frac{4}{3}(R+T)^3 + (R+T)^2L - (\frac{4}{3}R^3 + R^2L)\right]$ *Utility functions (preferences) for Volume and Weight Goals* $U_{Vol} = \begin{cases} 1.4\left(\frac{V}{V_{\text{target}}}\right) - 0.4\left(\frac{V}{V_{\text{target}}}\right)^2 & 0 < V < V_{\text{target}} \\ 0 & V = 0 \\ 1 & V \geq V_{\text{target}} \end{cases}$ $U_{Wgt} = \begin{cases} 1 - 0.6\left(\frac{W}{W_{\text{target}}}\right) - 0.4\left(\frac{W}{W_{\text{target}}}\right)^2 & 0 < W < W_{\text{target}} \\ 1 & W = 0 \\ 0 & W \geq W_{\text{target}} \end{cases}$

Utility-based Compromise DSP of pressure vessel design decision

$$U_{RangeVol} = \begin{cases} 1-0.6\left(\dfrac{V_{Range}}{V_{\text{RangeMax}}}\right)-0.4\left(\dfrac{V}{V_{\text{RangeMax}}}\right)^2 & 0 < V_{Range} < V_{\text{RangeMax}} \\ 1 & V_{Range} = 0 \\ 0 & V_{Range} \geq V_{\text{RangeMax}} \end{cases}$$

$$U_{RangeWgt} = \begin{cases} 1-0.6\left(\dfrac{W_{Range}}{W_{\text{RangeMax}}}\right)-0.4\left(\dfrac{W_{Range}}{W_{\text{RangeMax}}}\right)^2 & 0 < W_{Range} < W_{\text{Range}} \\ 1 & W_{Range} = 0 \\ 0 & W_{Range} \geq W_{\text{RangeMax}} \end{cases}$$

Find

System variables:

Radius (R)

Values of Deviation Variables:

d_{Vol}^{-}

d_{Wgt}^{-}

$d_{RangeVol}^{+}$

$d_{RangeWgt}^{+}$

Satisfy

System constraints:

$S_t - \left(\dfrac{PR}{T}\right) \geq 0$

$R - 5T \geq 0$

$(40 - R - T) \geq 0$

$(150 - L - 2R - 2T) \geq 0$

System Goals (Normalized):

$U_{Vol} + d_{Vol}^{-} - d_{Vol}^{+} = 1$

$U_{Wgt} + d_{Wgt}^{-} - d_{Wgt}^{+} = 1$

$U_{RangeVol} + d_{RangeVol}^{-} - d_{RangeVol}^{+} = 1$

$U_{RangeWgt} + d_{RangeWgt}^{-} - d_{RangeWgt}^{+} = 1$

Bounds on System Variables:

$0.1 \leq R \leq 36$

$0.1 \leq L \leq 140$

Utility-based Compromise DSP of pressure vessel design decision
$0.5 \le T \le 6$ *Non negative values of deviation functions* $d^-_{Vol}, d^+_{Vol} \ge 0$ $d^-_{Wgt}, d^+_{Wgt} \ge 0$ $d^+_{RangeVol}, d^-_{RangeVol} \ge 0$ $d^+_{RangeWgt}, d^-_{RangeWgt} \ge 0$ $d^+_{Vol}.d^-_{Vol} = 0$ $d^+_{Wgt}.d^-_{Wgt} = 0$ $d^-_{RangeVol}.d^+_{RangeVol} = 0$ $d^-_{RangeWgt}.d^+_{RangeWgt} = 0$ $k_{Wgt} + k_{Vol} + k_{RangeWgt} + k_{RangeVol} = 1$ **Minimize** *Deviation Function:* $Z = k_{Wgt}d^-_{Wgt} + k_{Vol}d^-_{Vol} + k_{RangeWgt}d^-_{RangeWgt} + k_{RangeVol}d^-_{RangeVol}$

4.4.1 Pressure Vessel Design under Imprecise Density

In this section, we consider the scenario where the density of the material used to manufacture the pressure vessel is imprecise, but strength of the material is known with certainty. The range of possible density values is known. With addition of more resources, the range of possible values can be reduced. For example, in the simplest material model, the range of predicted density values is between [0.003, 0.563] where the first number represents the lower bound and the second number represents the upper bound. On addition of more information in the material model, the range reduces to [0.010, 0.550]. The ranges of densities for different scenarios are plotted in Figure 4-16. The results of decision making using imprecise information are shown in Table 4-5. The table illustrates 10 levels of the material model, each row representing one imprecision level in the material model, associated with a range of achievable density values. These 10 corresponding scenarios of ranges of density, leading to different design decisions are labeled from 1 through 10 (in column ‘S. No.’); 1 corresponding to the most imprecise density information and 10 representing the least imprecise (in this example, the 10th level corresponds to a precise density value of 0.283).

Table 4-5 - Results from adding more information about material density

	Density		Decision	Utility at Decision Point	Value Metric (R1)		
S.No.	Lower Bound	Upper Bound	R	Expected Utility	Ex-Post Range	R1	R2
1	0.003	0.563	2.6	0.490	0.528	0.037	0.000
2	0.010	0.550	2.6	0.490	0.522	0.037	0.000
3	0.050	0.530	2.6	0.490	0.493	0.034	0.000
4	0.133	0.433	15.6	0.663	0.349	0.427	0.927
5	0.183	0.403	17.6	0.696	0.259	0.596	0.965
6	0.203	0.380	19.1	0.724	0.205	0.799	1.000
7	0.273	0.303	20.1	0.744	0.034	1.000	1.000
8	0.280	0.284	20.1	0.750	0.004	1.000	1.000
9	0.282	0.284	20.1	0.749	0.002	1.000	1.000
10	0.283	0.283	20.1	0.749	0.000	1.000	1.000

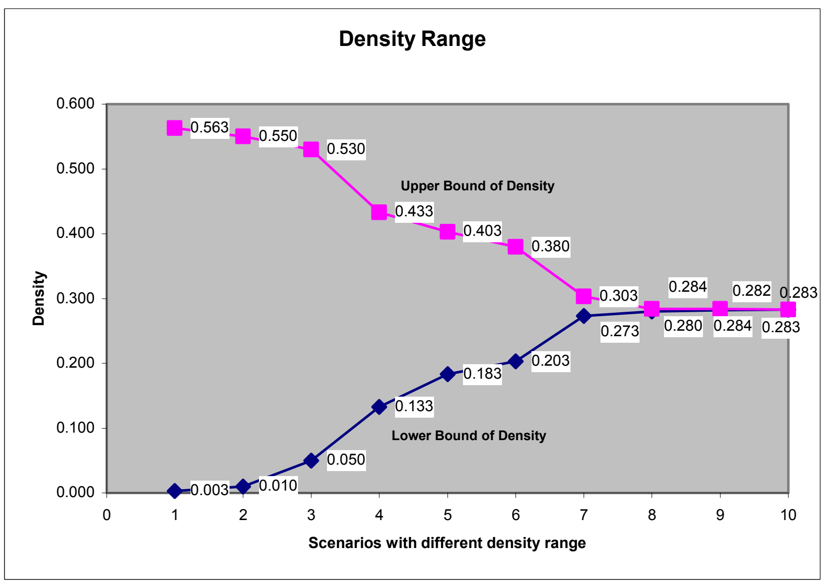


Figure 4-16 – Upper and lower bounds of density for different scenarios

This imprecision density information is used in the compromise DSP shown in Table 4-4 to select a value of radius that maximizes the objective function value given by a weighted sum of the average overall utility and the range of achievable utility values at the decision point. At each point the lower and upper bounds on utility, as well as the opportunity ratio (R1) and achievement ratio (R2) are shown in the table. Value of Information metrics including ex-post range, Ratios R1 and R2 are plotted in Figure 4-17 and Figure 4-18 for each of the imprecision scenarios. The achievement ratio is calculated for the cutoff utility value of 0.70.

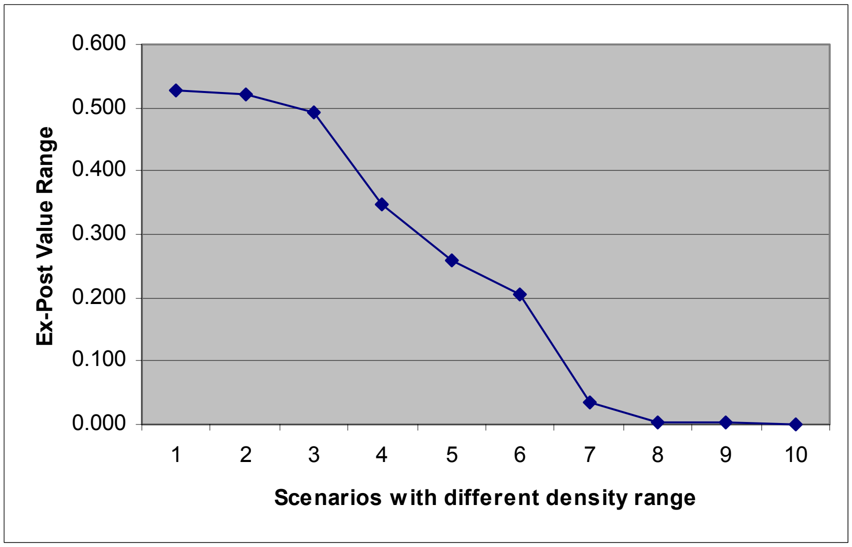


Figure 4-17 – Reduction in Ex-Post Value Range with addition of more information about density

As shown in the Figure 4-17, the ex-post value range reduces with addition of more information. After a certain stage, (point 7 in the figure), the ex-post range becomes almost zero. Hence, there is no benefit from addition of more information in the decision under consideration. It is also apparent from Figure 4-18, scenarios 1, 2, and 3 have values for ratios R1 and R2 close to 0. Hence, there is hope for getting a better solution (due to low R1) and the designer's wish is not achieved (due to low R2). Hence, there is a need for gathering more information about the density in order to achieve design objectives. A plot between design variable and utility values for scenario 1 is shown in Figure 4-19. In the figure, lower bound, upper bound, average value, and the objective function are plotted for various values of pressure vessel radius. The objective is labeled as *robust objective* because it is a combination of the average utility and the range of utilities. The maximum point in the objective function is highlighted with a square. It is shown that the maximum value of the objective function is for radius, R = 2.60.

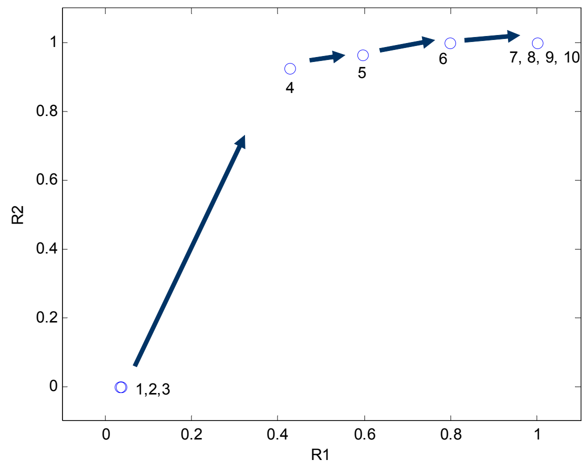


Figure 4-18 – Impact of addition of information on Opportunity Ratio and Achievement Ratio – imprecise density scenario

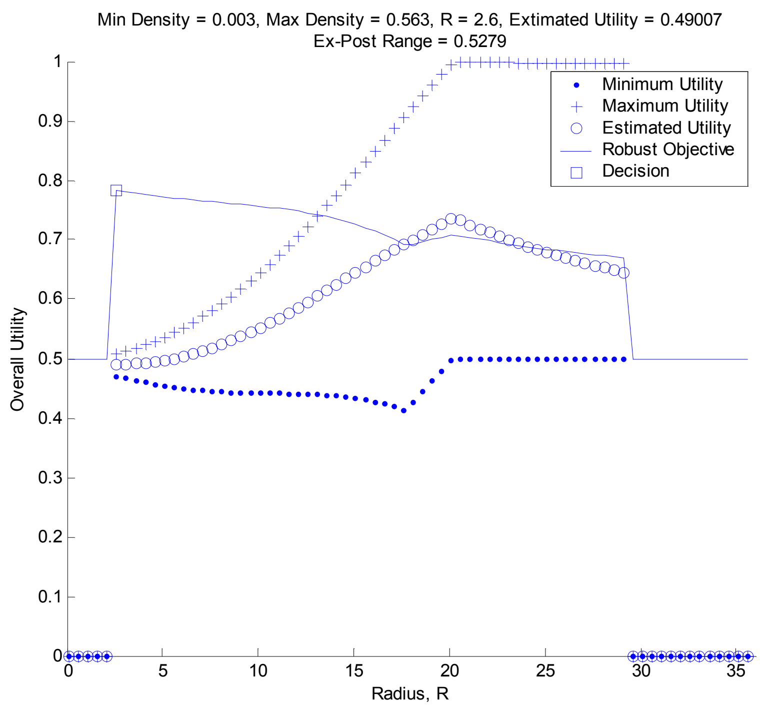


Figure 4-19 – Uncertain density - Scenario 1

The 4th and 5th scenarios in the imprecise material density information, as represented in Figure 4-18 lie in the region where the designer's wishes are better achieved but there is still hope of improving the design solution. The opportunity ratio value is 0.427 and the achievement ratio value is 0.926. The utility curves associated with scenario 4 are shown in Figure 4-20. The value of radius decided in this scenario is 15.6. In scenario 7, both the opportunity ratio and achievement ratio are equal to 1. This point reflects that there is little hope of achieving a better solution and the designer has achieved his/her desired solution. Hence, there is no need to gather more information to reduce the imprecision range of density value. The range of density corresponding to this scenario is [0.273 0.303]. The utility functions corresponding to this scenario are shown in Figure 4-21. Any reduction in this density range will result in the same values of achievement and opportunity ratio, which is apparent from the Scenarios 8 through 10 in Table 4-5. The utility functions corresponding to Scenario 9 are shown in Figure 4-22.

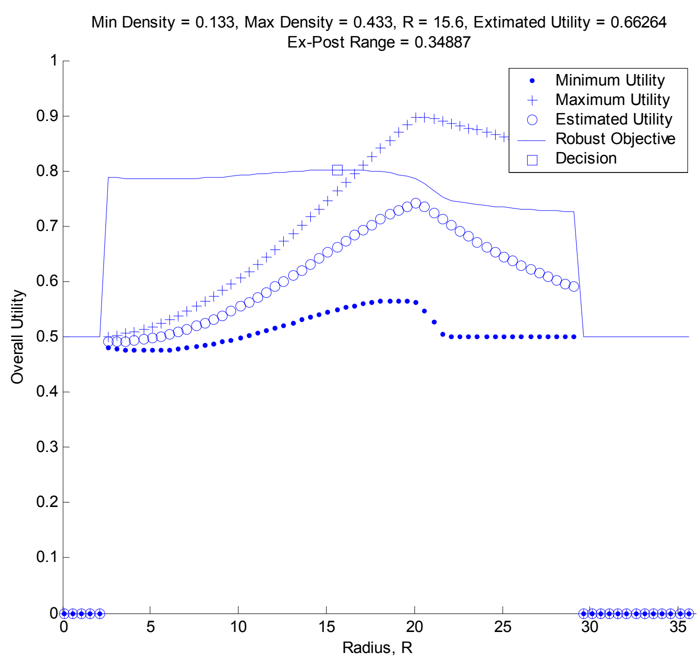


Figure 4-20 - Uncertain density - Scenario 4

Using this example, we show the application of value metrics – the achievement and opportunity ratios in making meta-level decisions such as decision about the need for addition of more information before making design decisions. In the following section, we present the scenario where density is known with certainty but the strength is imprecise.

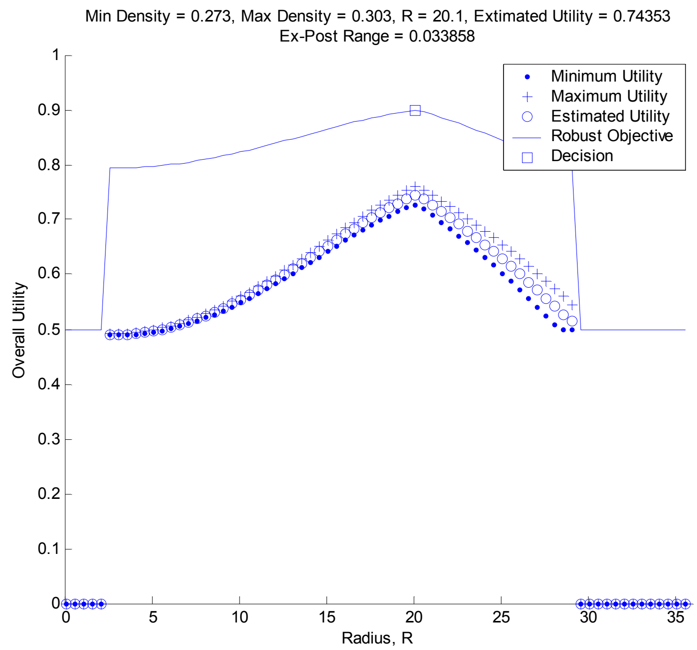


Figure 4-21 - Uncertain density - Scenario 7

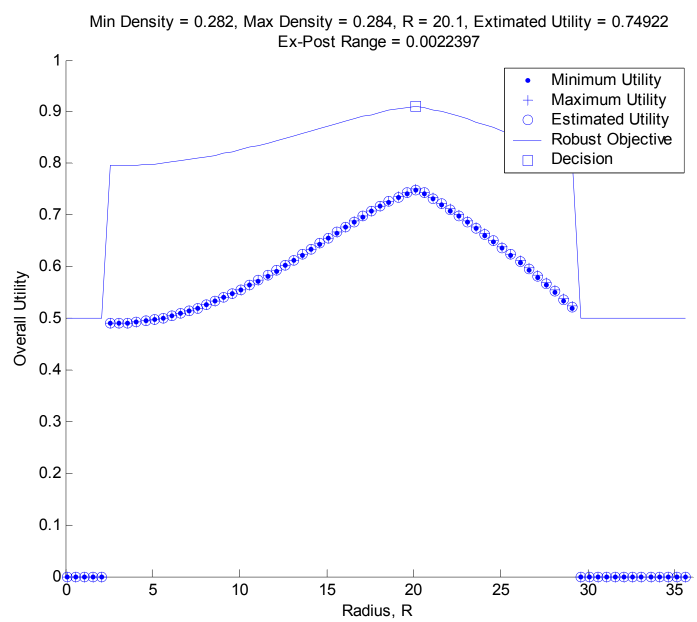


Figure 4-22 - Uncertain density - Scenario 9

4.4.2 Pressure Vessel Design under Imprecise Strength

Nine different scenarios for different levels of imprecision in strength are considered in this section. The ranges of strength values for different scenarios are shown in Figure 4-23. Decisions about the numerical values of radius are shown for each scenario in Table 4-6. The ex-post values for each level are plotted in Figure 4-24. Corresponding values of opportunity and achievement ratios are shown in Figure 4-25.

Table 4-6 - Results from adding more information about material density

	Strength		Decision		Value Metric (R1)		
S.No.	Lower Bound	Upper Bound	R	Expected Utility	Ex-Post Range	R1	R2
1	50000	650000	6.1	0.505	0.245	0.000	0.000
2	100000	600000	12.6	0.602	0.147	0.000	0.000
3	120000	600000	15.1	0.655	0.094	0.000	0.000
4	140000	560000	17.6	0.708	0.042	0.000	1.000
5	160000	540000	20.1	0.749	0.000	1.000	1.000
6	200000	500000	20.1	0.749	0.000	1.000	1.000
7	250000	450000	20.1	0.749	0.000	1.000	1.000
8	300000	400000	20.1	0.749	0.000	1.000	1.000
9	350000	350000	20.1	0.749	0.000	1.000	1.000

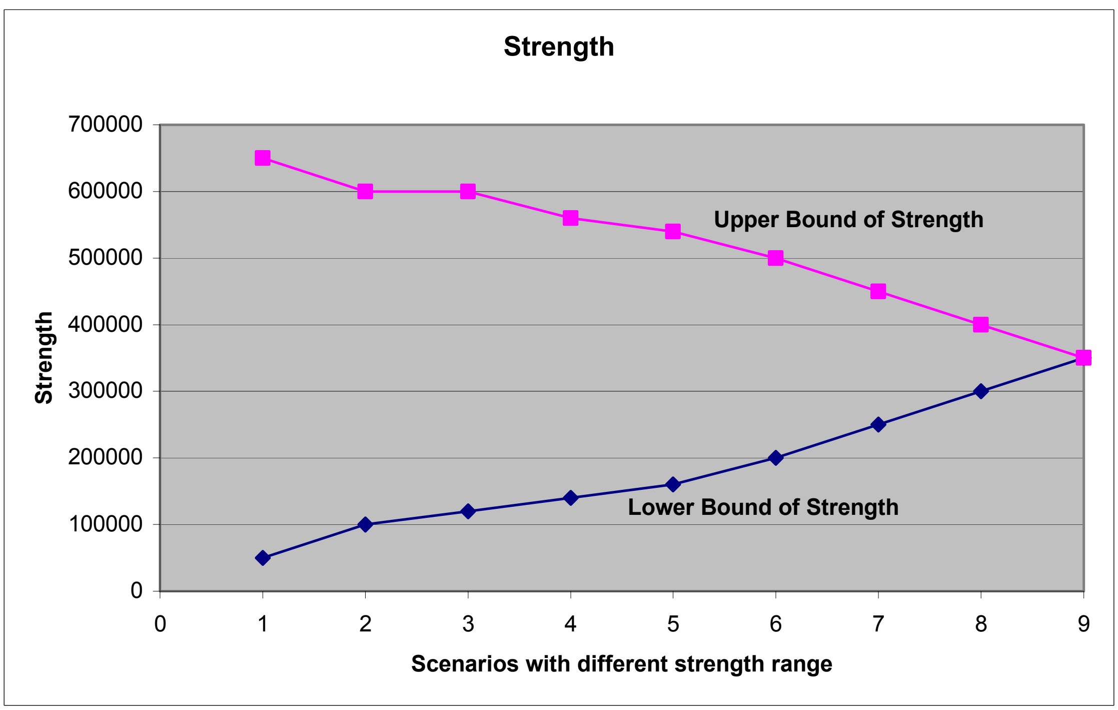


Figure 4-23 – Upper and lower bounds of strength for different scenarios

In the first scenario, the imprecision interval for strength is [50000 650000]. This scenario corresponds to the utility functions plotted in Figure 4-26. The value of both R1 and R2 is 0.0, which shows that there is a need for gathering more information and reducing the range of imprecision interval of strength. Same is the scenario with scenarios 2 and 3. In scenario 4, the opportunity ratio is 0.0, which represents that there is a possibility for improving the design solution. The achievement ratio is 1.0, implying that designer's objective is met. In Scenario 5, both the ratios equal 1.0, which is an indication that the designer's objective is met and there is no possibility of improvement in the design solution. Hence, the designer may stop reducing the imprecision interval because after this stage, any reduction in the range for strength does not improve designer's decision making capability. This is evident from the fact that decision made in scenarios 6 through 9 is the same (Radius = 20.1). The utility function values for Scenarios 3, 5, and 7 are shown for illustration purposes in Figure 4-27, Figure 4-28, and Figure 4-29 respectively.

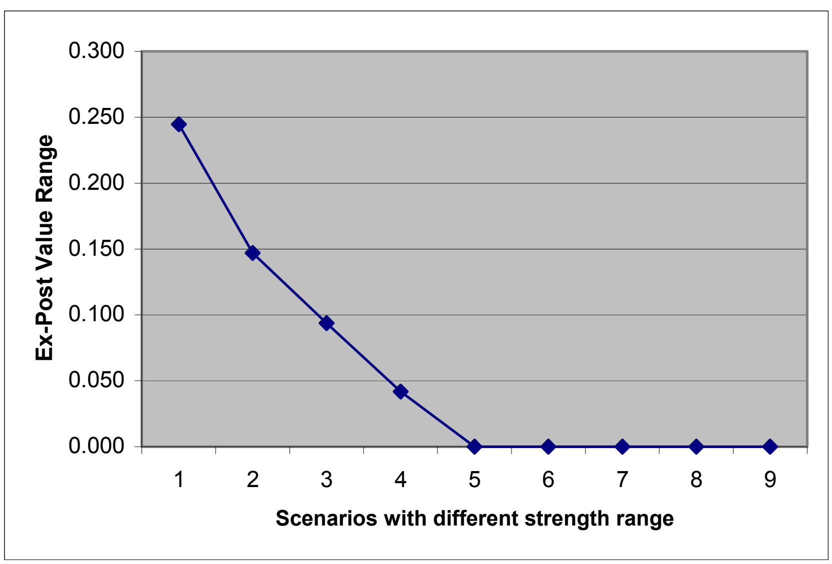


Figure 4-24 - Reduction in Ex-Post Value Range with addition of more information about density

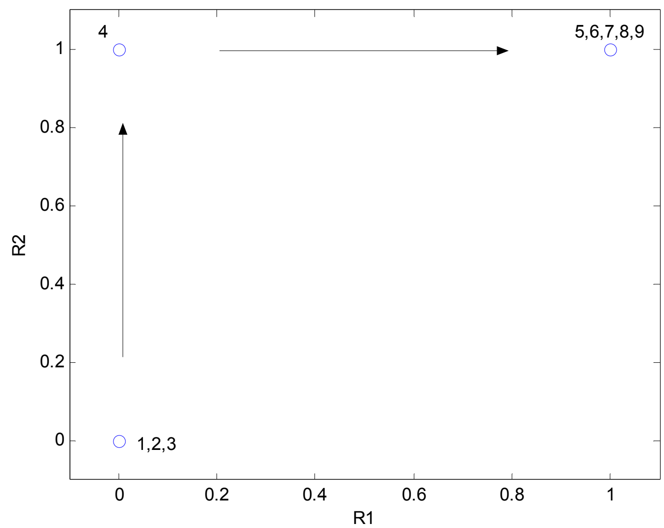


Figure 4-25 - Impact of addition of information on Opportunity Ratio and Achievement Ratio – imprecise strength scenario

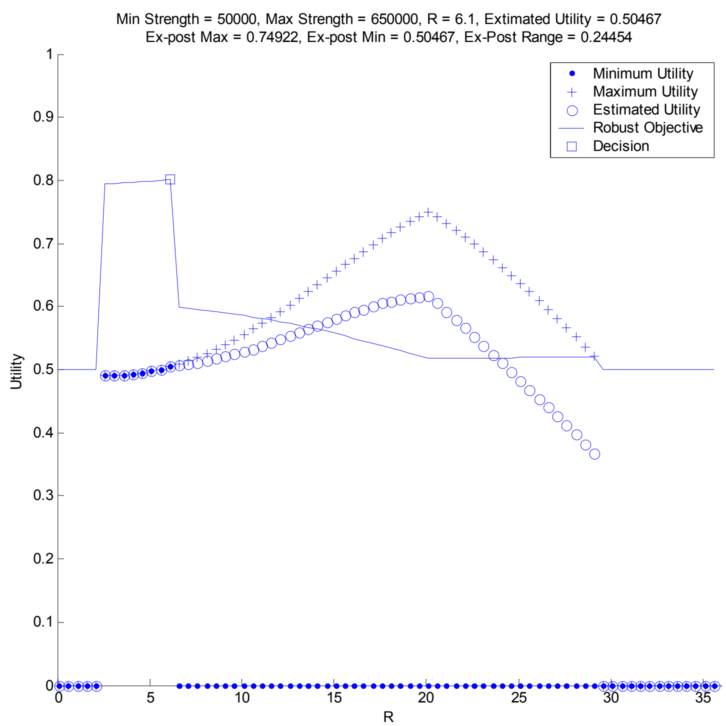


Figure 4-26 – Uncertain strength - Scenario 1

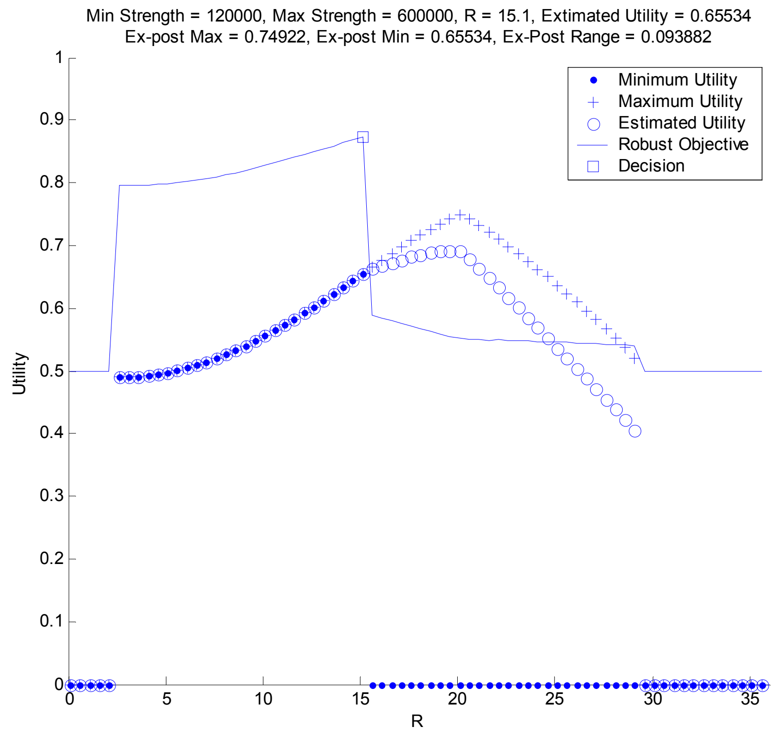


Figure 4-27 - Uncertain strength - Scenario 3

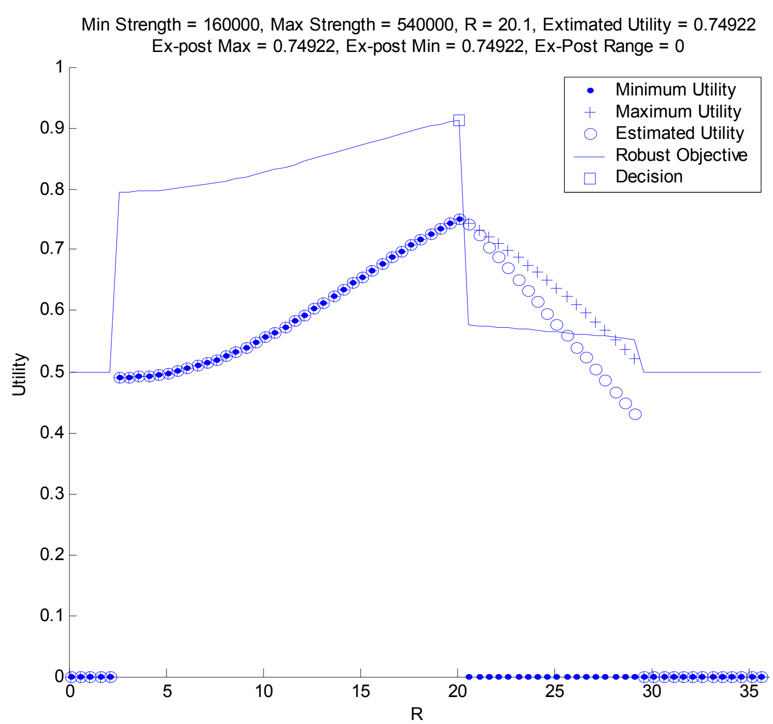


Figure 4-28 - Uncertain strength - Scenario 5

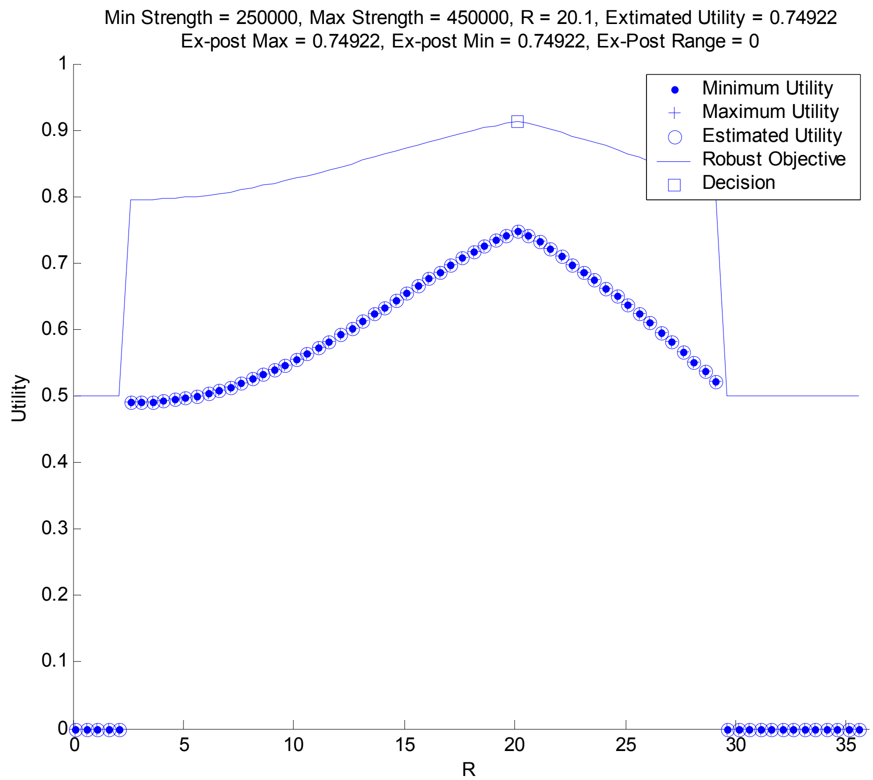


Figure 4-29 - Uncertain strength - Scenario 7

4.5 On Verification and Validation

In this chapter, three aspects of the validation square are addressed - theoretical structural validation, empirical structural validation, and empirical performance validation. The details of the validation square as addressed in this chapter are discussed in Sections 4.5.1, 4.5.2, and 4.5.3. An overview of the validation performed in this dissertation is presented in Figure 1-13. Specific details related to the validation of value of information are presented in Figure 4-30.

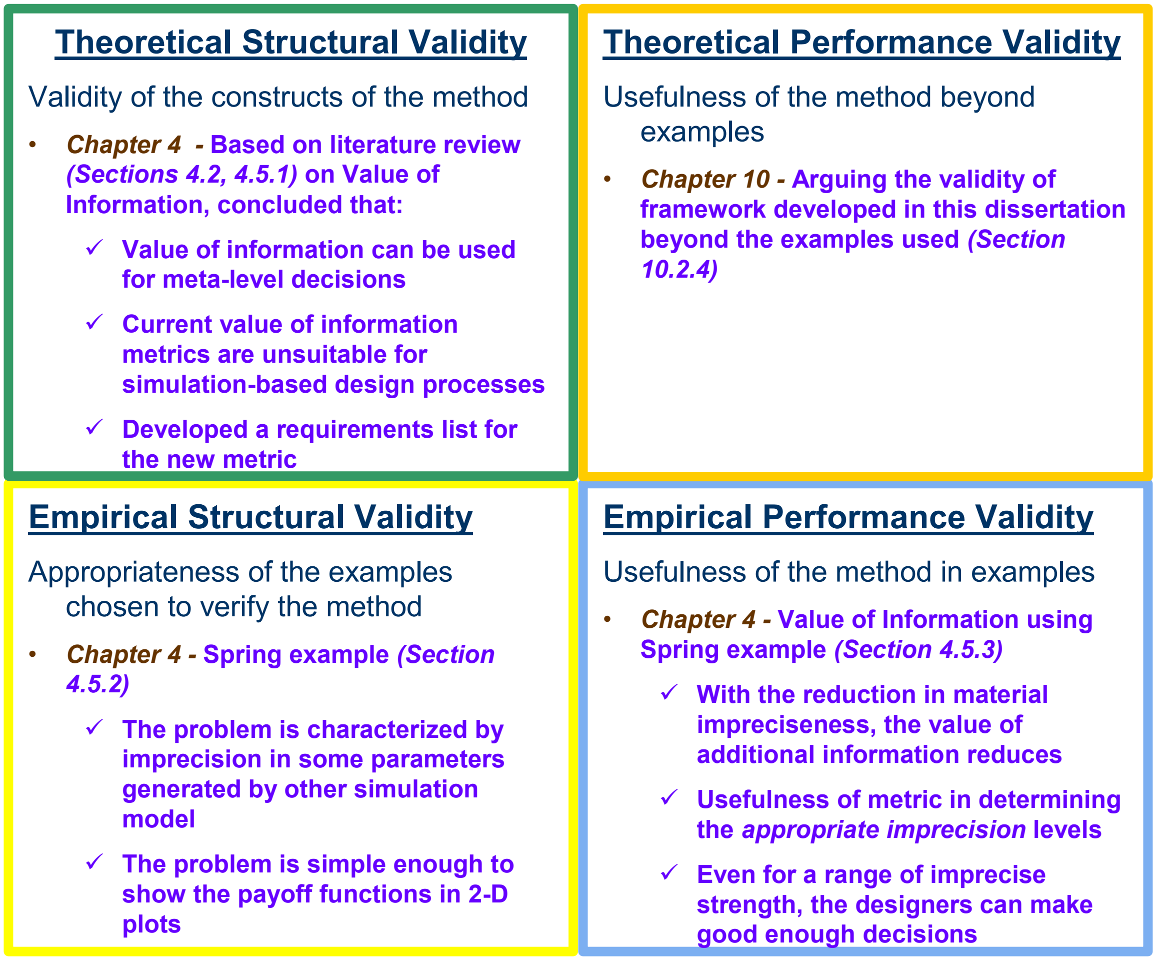


Figure 4-30 – Summary of validation of value of information metric addressed in Chapter 4

4.5.1 Theoretical Structural Validation

Theoretical structural validation refers to accepting the validity of individual constructs used in the method and accepting the internal consistency of the way the constructs are put together. The internal consistency of the individual constructs can be checked by a critical review of the literature. In this chapter, one construct is used – value of information for decision making. In the Section 4.1, we have argued why value of information is appropriate as a metric for making meta-level decisions. Different design decisions such as determining the right level of refinement of simulation models and the right level of simplification of design processes can be considered as meta-design decisions, where the designers need to determine whether additional information is necessary or not. Based on the existing literature, it is shown that value of information has been previously used for making similar decisions on addition of information for decision making. However, from the critical review of the literature, it is identified that the current metrics for value of information are unsuitable for decisions related to simulation-based design processes. This is primarily due to the assumption that additional information that can be gathered in a decision making process provides better knowledge about the probability of events. Hence, only statistical uncertainty is captured in the existing metrics. Other limitations are due to the consideration of expected value of payoff from the decision making process. The deviation from the expected value is not considered in the decision making process. From the limitations of existing metrics, three requirements for a metric suitable for meta-design decisions are listed in Section 4.2. These requirements are then embodied in a new value of information metric that consists of three components, each indicating a different aspect of the impact of added information. Due to this logical procedure of literature review, gap analysis, and development of new metric based on the requirements, the theoretical structural validity of the individual constructs is accepted. The second step in the theoretical structural validity – accepting internal consistency of the way constructs are put together is not required in this chapter because only one construct is used.

4.5.2 Empirical Structural Validation

The empirical structural validation involves accepting the appropriateness of the example problems used to verify the performance of the method. In this chapter the performance of the value of information metric is measured by using a pressure vessel design problem. A designer employs models for determining the weight and volume of the pressure vessel, which are then used for making decisions about the dimensions of the pressure vessel. These models utilize information about the material properties (that can either come from experimental data or from a material behavior simulation model). Hence, the important characteristic of the problem is imprecision in some of the parameters that are generated by another simulation model. This is a typical scenario in a simulation-based multiscale design problem. Imprecision in the simulation models is one of the key requirements for a new value of information metric. The step-wise reduction of imprecision in the input parameters, shown in the example in Section 4.4, represents the refinement of simulation models that generate information about these parameters and is also common in multiscale design. Further the problem is simple enough due to *a)* the simple simulation models for evaluating the pressure vessel's weight and volume, and *b)* a single design variable, which enables the visualization of results (overall utility and its lower and upper bounds) on a two dimensional plot. Hence, the problem is appropriate for demonstrating the usefulness of value of information metric in making design process related decisions. Note that in this chapter, we are only testing the performance of the metric in the presence of imprecision that is modeled with lower and upper bounds on parameter values that can be mapped to lower and upper bounds on overall payoff. The pressure vessel design example does not include consideration for both imprecision (range based uncertainty) and variability (statistical uncertainty). The consideration of both variability and imprecision is validated through the materials design example in Chapter 9.

4.5.3 Empirical Performance Validation

Empirical performance validation consists of two steps – accepting the usefulness of the outcome with respect to the initial purpose and accepting that the achieved usefulness is related to applying the method. In this chapter, we show that the application of value of information metrics helps designers in making meta-level decisions such as determining the right level of refinement of simulation models. The specific example considered here contains imprecision in two parameters – strength and density. The results presented in Sections 4.4.1 and 4.4.2 show that the upper bound on ex-post utility decreases with the addition of more information (i.e., reduction of range on imprecise parameters). With the reduction in the range of density, the range on the utility also reduces in size. This results in reduction of the upper bound on ex-post utility. With the reduction in the range for strength, a similar trend is observed in the upper bound on ex-post utility. However, the key difference between the trends observed for density and strength is that by reduction in the range for density, the value of ex-post range continuously reduces and drops to zero only when the imprecision in density is completely eliminated. Whereas, on reducing the range for strength, the upper bound on ex-post value drops to zero even when there is imprecision in strength. This is intuitive because the strength is associated with a constraint in the designer's decision. If all the values of strength in the imprecision range are such that they satisfy the strength constraint, the exact numerical value of strength does not affect the decision. Hence, the value of additional information (via reducing the range of strength) is zero. This can also be interpreted as follows – if the worst case scenario of strength value satisfies the strength constraint, then the value of information is zero. Hence, the results by using the value of information are acceptable and inline with the designers' expectations. Based on this argument, we assert that the empirical performance validity is achieved. Since the empirical performance validity is based mainly on the scope of the method and the type of problem considered, it is important to consider scenarios where the metric would not work. Hence, the next section (4.5.4) is devoted to analyzing the limitations of the metric and possible avenues for future work.

4.5.4 Limitations of Value of Information Metric and Opportunity for Future Work

The value of information metric presented in this chapter (Section 4.3) is based on the assumption that information about the lower and upper bounds on the imprecise variables is available. Hence, different levels of refinement of the simulation models and different design processes need to be characterized with the information about possible lower and upper bounds of outputs. The limitations of the metric are discussed in the following:

1. The metric is developed by only considering the information about the improvement in payoff resulting from the decision. It does not include the cost of gathering the additional information (the cost of reducing the range of imprecise variables in the case of simulation model refinement). It is assumed that for a given step in the series of refinement steps, the designer evaluates the estimated cost of gathering information and the value of this added information. Using these two indicators, the designer makes the decision on whether additional information is worth gaining. Another option of including the cost information is to include it directly in the payoff calculation.
2. Since the metric is based on the availability of lower and upper bounds, it cannot be used if this information is not available. For example, if there are different fidelities of simulation models available but there is no information about the

bounds within which the actual behavior lies, then the metric cannot be used. The metrics need to be modified in the future to include this information.

3. The metric can only be used to determine whether a given level of refinement of simulation model is appropriate for making a particular decision or not. It does not help designers in determine the level (extent) of refinement of simulation model.
4. A simulation model may have different modes of refinement. The value of information metric, as shown in his chapter, does not advise the designers how to refine the simulation models – which mode of refinement should be used in a given scenario.
5. The value of information metric provides a conservative estimate of the possible improvement in the payoff because the metric is based on the upper bound of difference between payoff obtained and payoff that could be obtained by gathering more information.
6. The value of information metric is useful for comparing simulation models and design processes that are improvements or simplifications of each other in the context of a given decision. In other words, the metric can only be used for sequentially improving the simulation models. The designers need to employ the models and processes to make decisions. It cannot be used for directly selecting from a list of available models, without employing any of the models.

4.6 Role of Chapter 4 in This Dissertation

The relationship of the value of information metric developed in this chapter with the other chapters is shown in Figure 4-31.

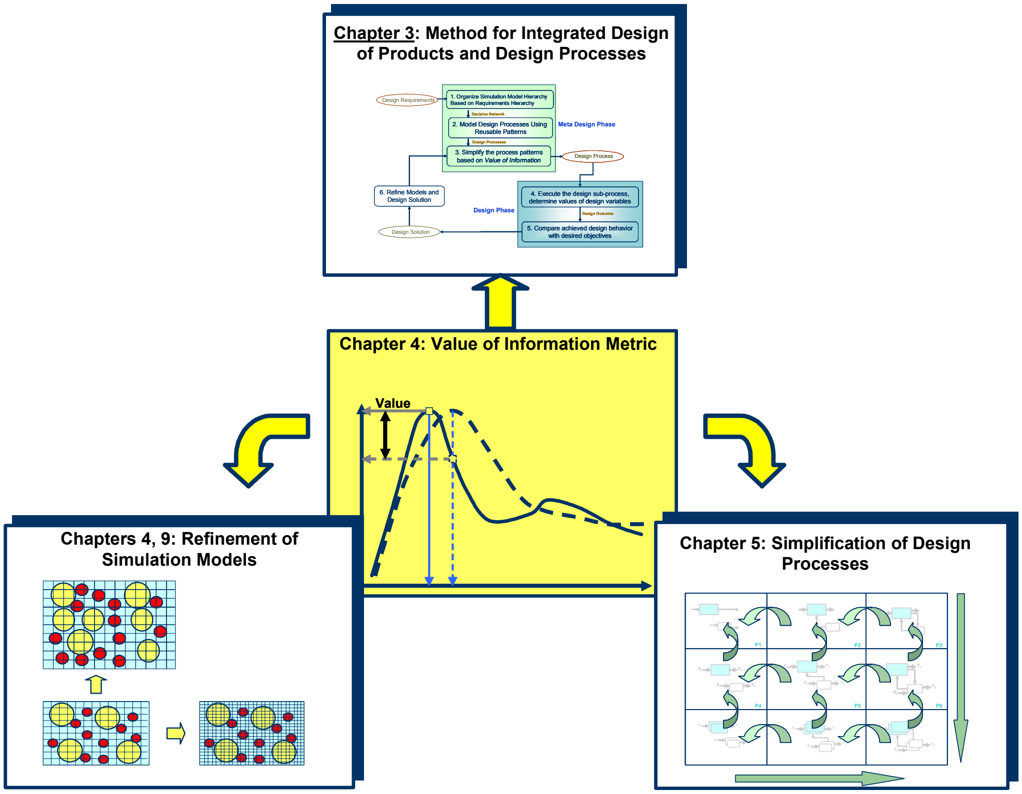


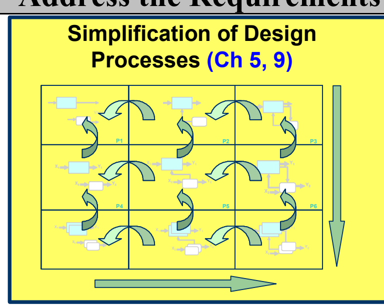
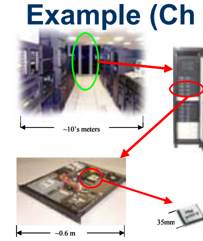
Figure 4-31 – Relationship of Chapter 4 with other chapters in the dissertation

The focus of this chapter is on developing the value of information metric. This metric is an embodiment of the Hypothesis H2.1. The metric is used in this chapter for determining the right level of refinement of simulation model. The metric is based on imprecision and variability. The metric is used in the Steps 3 and 6 of the design method presented in the previous chapter (see Chapter 3). The metric developed in this chapter is an essential component for answering the Research Question 2. It is used in the following chapter (Chapter 5) for determining the appropriate level of simplification of design process.

Chapter 5 Design Process Simplification Using Value of Information Metric

In this chapter, we address the fourth requirement of the design framework, which involves the provision of *support for simplification of complex design processes without affecting the product performance*. The requirement, component of the framework, and corresponding validation example are listed in Table 5-1. This table is a subset of the list presented in Table 1-6. The framework component developed in this chapter to address the fourth requirement comprises of two methods for simplification of design processes via scale decoupling and decision decoupling respectively. These methods are used to answer the second research question through the embodiment of hypothesis H2.2: *design processes can be simplified using decoupling of scales, decisions and functionalities*. The details of the research question and the role of this chapter in the overall framework development and validation are discussed in Section 5.1.

Table 5-1 - The requirement and component of the framework for integrated design of products and design processes addressed in Chapter 5

Framework Requirements	Components of the Framework Developed to Address the Requirements	Validation Examples
4) Support simplification of complex design processes without affecting the performance of the product		**Datacenter Design Example (Ch 5)**  ***Purpose:*** To validate the use of value-of-information based metrics for design process simplification

5.1 Frame of Reference – Answering the Research Question 2 (Simplification of Design Processes)

Everything should be made as simple as possible, but not simpler. This is a famous quote by Albert E. Einstein. The quote is particularly important in the context of design, where the objective is to make satisficing decisions. While complex design processes that consider all interactions lead to better designs, simpler design processes, where some interactions are ignored, are faster. The right level of simplification of design processes is the one that reduces the design effort significantly without a major negative impact on the quality of decisions. Hence, designers are faced with the following question – "what is the right level of simplification of design processes?" In Chapter 4, a metric for making such meta-level decisions is presented. It can be used to quantify the impact of addition or removal information on designers' decisions. It is applied to selecting the right level of refinement of simulation models. In this chapter, we utilize the value of information metric developed in Chapter 4 for making design process simplification decisions. Although design processes can be simplified in variety of ways, such as minimizing iterations, minimizing the information flow between teams, etc., the scope of this chapter is limited to simplifying the basic design process building bocks defined in terms of the nine interaction patterns. As discussed in Chapter 3, complete design processes can be modeled using hierarchical combinations of design process building blocks. Hence, if the interaction patterns can be simplified using the value of information metric, complete design processes can also be simplified successively using the same metric and by following the same series of steps. The focus in this chapter is on answering the second research question for the dissertation – "How should multiscale *design processes be systematically simplified* and models refined in a targeted manner to support quick design decision making without compromising their quality? The simplification of design processes' aspect of this research question is addressed in this chapter. The hypothesis used to answer this research question is that design processes can be simplified using decoupling of scales, decisions, and functionalities. The first two aspects of this hypothesis – scale and decision decoupling are used in this chapter to perform design process simplification.

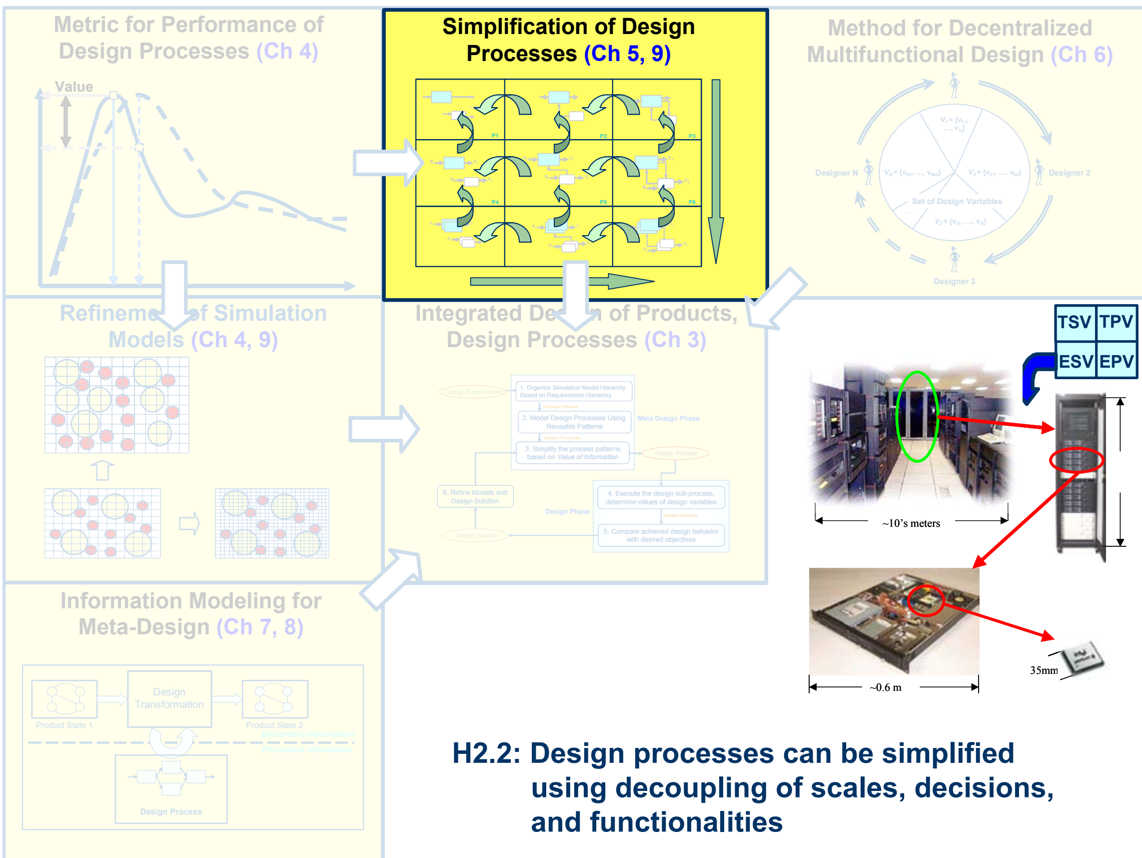


Figure 5-1 – Hypothesis addressed and validation example used in Chapter 5

The basic concepts used for developing the method for simplification are discussed in Section 5.2. The focus in Section 5.3 is on scale decoupling. The method for scale decoupling is presented in Section 5.3.1 and an example for validating the method is presented in Section 5.3.2. The focus of Section 5.4 is on decision decoupling. The method for decision decoupling is presented in Section 5.4.1 and an example problem to validate the method is presented in Section 5.4.2. The example problem used in this chapter is the design of a datacenter cooling system, which is introduced in Section 5.3.2. Finally, verification and validation is discussed in Section 5.5. The research hypotheses addressed in this chapter are highlighted in Figure 5-1. The validation example used in this chapter is that of a datacenter cooling system design.

5.2 Elements of the Proposed Simplification Strategy for Simulation-based Multiscale Design Processes

The approach for simplification of design processes presented in this chapter is based on the following four constructs: *a) interval arithmetic* to model imprecision, *b) robustness* to make decisions in the presence of variability and imprecision, *c) value of information* based metric to determine the impact of design process simplification on designers' decision making capabilities, and *d)* representation of design processes using commonly occurring design *process patterns* modeled in terms of the interactions between process elements. The use of each of these four fundamentals is discussed in the following sections.

5.2.1 Intervals for Simplification

The strategy for simplification adopted in this chapter is to ignore interactions between design process elements that do not have a significant effect on design decisions. For example, consider a system whose characteristic is given by the following mathematical relationship:

$$Y = F(X)$$

where, X is a vector of input variables $\{x_1, x_2, ..., x_n\}$ and Y is a set of outputs $\{y_1, y_2, ... y_m\}$. The system is shown in Figure 5-2. Although the overall system response (Y's) are dependent on all the system inputs, the system can equally well be represented as two sub-systems corresponding to the response subsets Y_A and Y_B that depend on X_A and X_B respectively. However, due to the interactions between these two subsystems, individual responses Y_A and Y_B cannot be entirely reproduced by X_A and X_B, causing an error in each of the subsystem responses (see Figure 5-2). If this error has a negligible or no impact on the designer's decision making capability, the system can be simplified into two subsystems that are independent of each other.

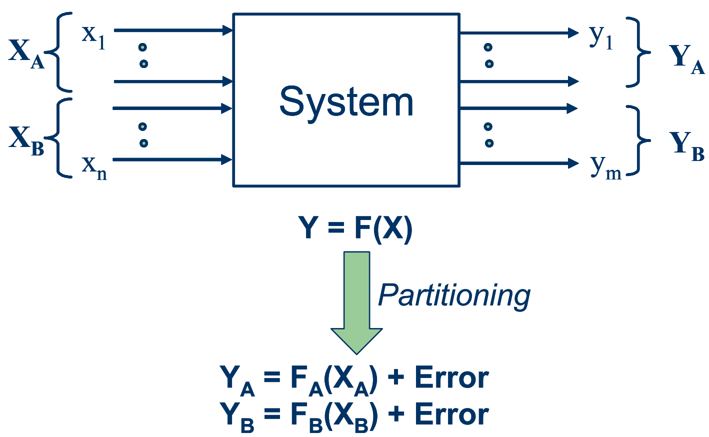


Figure 5-2 - Partitioning as a type of simplification

In order to capture the effect of simplification of interactions between different subsystems, we replace the information flow that is ignored in a subsystem with an interval representing the possible values that can be assumed by the information link. For example, in the partitioned Subsystem 1 shown in Figure 5-3, Y_A is the response and X_A is the input. X_B is the result of interaction with subsystem 2, which is replaced with an interval $[X_{B,\min}, X_{B,\max}]$. This range of information accounts for the imprecision introduced in the system due to simplification of interactions between subsystems, and results in a range of values for the outputs Y_A, denoted by $[Y_{A,\min}, Y_{A,\max}]$. The width of the output interval $[Y_{A,\min}, Y_{A,\max}]$ increases with increase in the width of input interval $[X_{B,\min}, X_{B,\max}]$, whereas the range reduces with the reduction of imprecision caused by simplification.

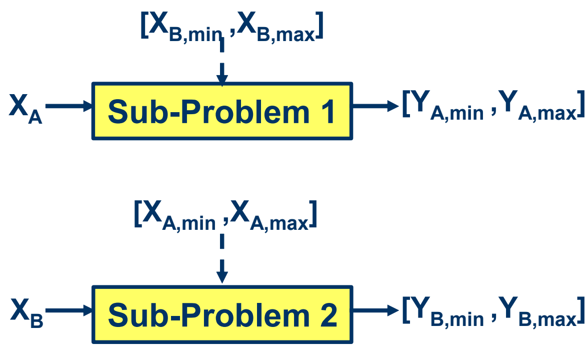


Figure 5-3 – System partitioning using intervals

In this dissertation, we use the intervals to model the imprecision introduced in a system due to simplification. Using intervals, we can evaluate the range of output values if the range of input values is given. These output values from the system analysis correspond to system performance. The system performance is subsequently related to designers' preferences through utility functions. Due to the ranges in input variables (resulting from simplification) there is a range of achievable system performance, which in turn results in a range of utility values. A designer makes decisions about the values of design variables using information about achievable range of utility values.

5.2.2 Robustness for Making Decisions under Simplification Induced Imprecision

The idea of robustness is used to make decisions under uncertainty. Uncertainty is important in this discussion on simplification and results from two sources – *a)* imprecision introduced due to simplification and *b)* system uncertainty due to noise factors. The general robust design approach is illustrated in Figure 5-4, where the objective is to decide on the values of design variables (also called control factors) that satisfy the desired values of response variables, while also minimizing the impact of changes in noise factors on the responses. This is achieved through different goal formulations in the design decisions. The approach adopted in this dissertation is to model two goals associated with each response – a) achievement of the desired target value for each response, and b) minimization of deviation of response due to changes in the noise factor. Designers assign preferences to both these goals in the form of utility functions and the overall utility function is evaluated for different points in the design space. The point in the design space that maximizes this overall utility function is selected by the designer.

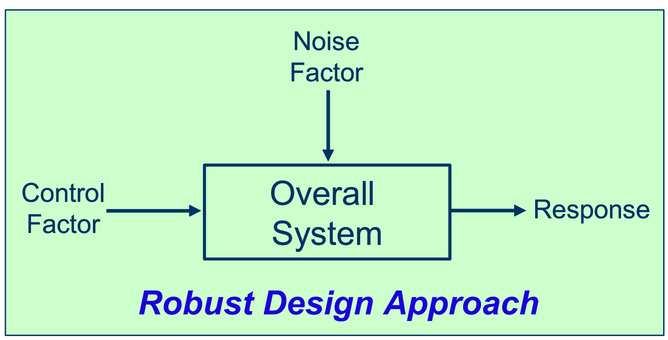


Figure 5-4 - General robust design approach

The concept of robustness has been used in design literature in various different uncertainty scenarios such as uncertainty in noise variables, design variables, uncertainty in simulation models, and uncertainty in a chain of models. The existing literature on these types of robust design is discussed in Section 2.3. The distinguishing feature of robustness considered in this chapter is that we use the concept of robustness to design systems that are also insensitive to simplification assumptions in the design processes.

5.2.3 Value of Information

Different levels of simplification result in different design decisions. The selection of a right level of simplification in the design process depends on the effect on designers' decision making capability. Increasing the level of simplification of a design process increases imprecision, thereby reducing the designer's capability to make good decisions. This effect of simplification of a design process on a designer's decision making capability is quantified by the value of information metric proposed in Chapter 4. The metric developed in Chapter 4 is based on determining the difference between the overall system performances based on the decisions made with and without imprecision due to simplification. It quantifies the effect of both the extent to which designer's objectives are satisfied and the possibility of improving the design solution by relaxing simplifications. The metric requires the knowledge of ranges of achievable overall utility values for different points in the design space. This information is available because the ranges of values assumed by interactions is ignored during the simplification process are known. The value of information at a particular simplification level of the design process determines whether there is a need to relax some simplification assumptions or not. If at a given simplification level, the value of information shows that the designers' objectives are met and the potential benefits by adding more details to the system is not likely to improve the design decision, the simplified design process can be used for designing.

5.2.4 Interaction Patterns for Modeling Design Processes

At any level of abstraction, design processes can be broken down into patterns that repeat themselves. Using the idea of *patterns*, any complex network can be broken down into a common set of patterns at any level of abstraction. The idea of patterns in design processes is discussed earlier in Chapter 3. Since the hypothesis is that the same patterns occur at different levels of details of the design processes, it is required to understand the simplification of interaction patterns only. We are only considering the interaction patterns that consist of two components interacting with each other. The same principles extend to process elements where more than two components interact. The interactions patterns considered in this research are shown in Figure 5-5. These interaction patterns are described in terms of a matrix with three rows and three columns and are first introduced in Section 3.5.2 and illustrated in Figure 3-11. The rows of the matrix are: *a)* information flow between simulation models, *b)* information flow between decisions, and *c)* multifunctional design. The columns of the matrix describe the level of interaction between the components and are described as *a)* independent interactions, *b)* dependent interactions, and *c)* coupled interactions. The interaction patterns are marked with labels – P1 through P9. Based on these nine interaction patters, three types of simplification are considered in this research. These include scale decoupling, decision decoupling, and functional decoupling.

Scale decoupling refers to simplification of interactions from pattern P3 to pattern P2 and from pattern P2 to Pattern P1. It refers to the simplification of information flow between two simulation models used for making a single decision. In the scale simplification (decoupling) scenario, there is a single set of design variables and a single set of objectives. The information needed to make that decision is generated from two separate simulation codes (generally at different scales) that may need to be executed in a coupled fashion. The task in scale simplification is to determine whether the coupled nature of simulation models (pattern P3) is important for making the decision or it can be simplified into a sequential information flow (pattern P2) or into an independent execution (pattern P1). Scale decoupling is addressed in detail in Section 5.3.

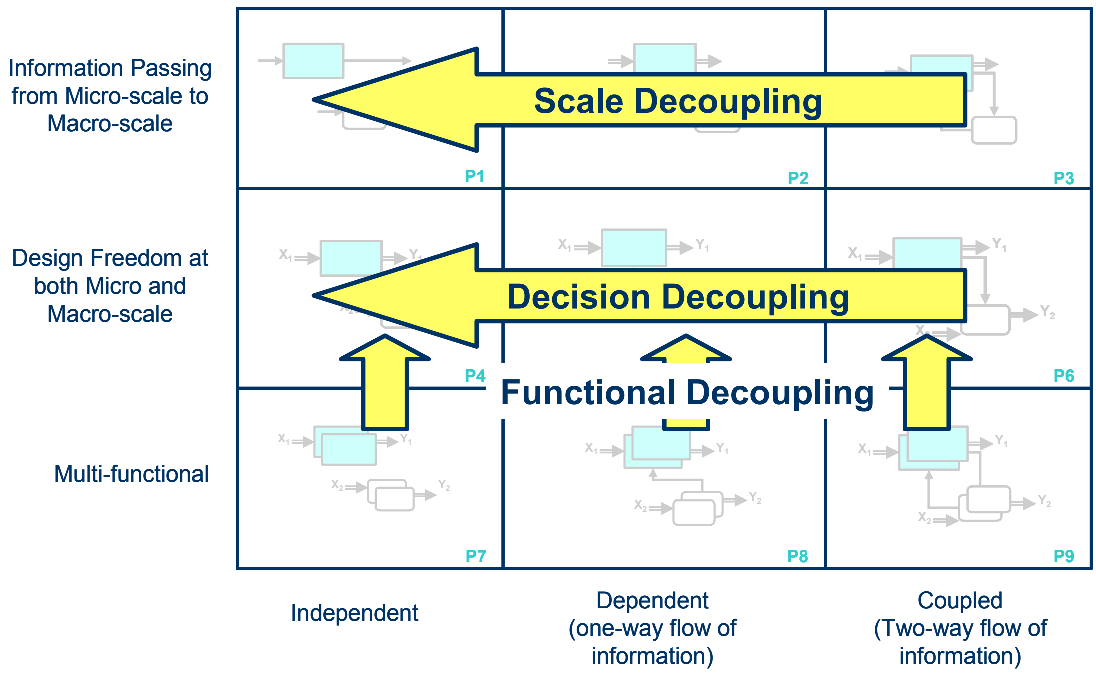


Figure 5-5 – Types of simplification considered in this dissertation – scale decoupling, decision decoupling, and functional decoupling

Decision decoupling refers to simplification of interaction patterns from pattern P6 to pattern P5 and pattern P5 to pattern P4. It refers to the simplification of information flow between decisions from a coupled decision making to an independent decision making. A decision decoupling scenario is characterized by multiple decisions – each associated with a set of design variables that need to be decided upon. Each of the set of design variables affects a common set of objectives. The task in decision decoupling is to determine an appropriate interaction level between the decisions such that the design objectives are satisfied with the minimum complexity in the design process. Decision decoupling is discussed in greater detail in Section 5.4.

Functional decoupling refers to the simplification from pattern P9 to pattern P6, pattern P8 to P5, and pattern P7 to P4. This is important in the case of multifunctional design where the product is designed to satisfy more than one functional requirement that drive the design into different directions. Such design scenarios are characterized by multiple sets of design variables (possibly overlapping), whose values can be selected for satisfying multiple objectives. The task of functional decoupling is to determine which functional requirements can be satisfied independently and which of those should be designed for in a concurrent fashion. Functional decoupling also depends on how the design variables are partitioned for satisfying different functional requirements. Hence, the task in functional decoupling is also to determine the appropriate design space partitioning, the details of which are discussed in Chapter 6. A summary of the characteristics of three types of simplification discussed in the following sections is provided in Table 5-2.

Table 5-2 - Characteristics of three types of simplifications considered in this chapter

Section 5.3	*Scale Decoupling:* Single set of design variables, single set of objectives
Section 5.4	*Decision Decoupling:* Multiple sets (corresponding to different scales) of design variables, single set of objectives
Chapter 6	*Functional Decoupling:* Multiple sets of design variables, multiple set of objectives (corresponding to multifunctional design)

The general steps in decision and scale decoupling are listed in Figure 5-6 and are discussed next. Using the four constructs described in this discussion, the general steps for scale, decision and functional decoupling include:

1. Identify the interaction pattern (P1 – P9) and the type of decoupling to be considered

2. Select a simplest interaction pattern corresponding to the decoupling to be considered (P1 in the case of scale decoupling, P4 in the case of decision decoupling, etc.)

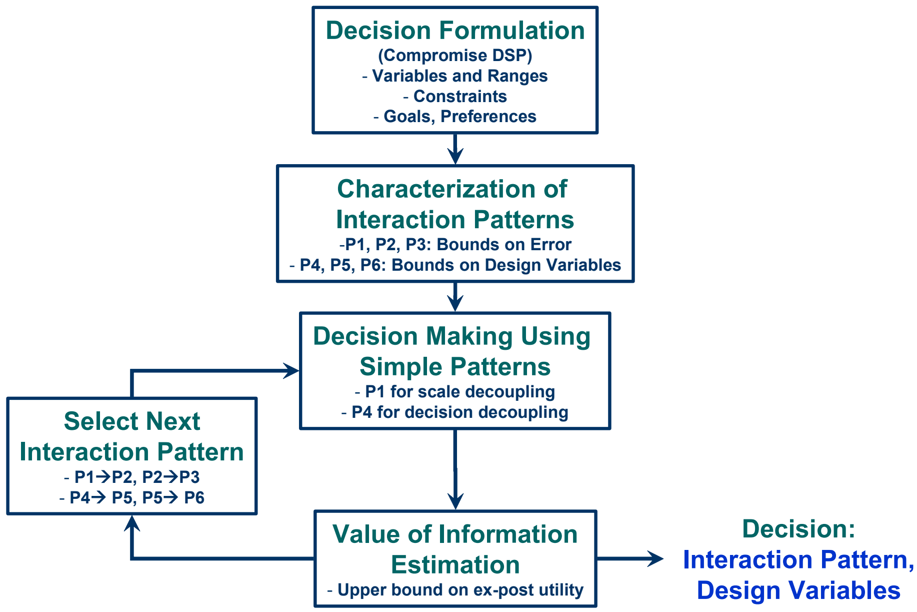


Figure 5-6 – General Steps for Decision and Scale Decoupling

3. Make a decision about design variables using the simplest model and determine the range of overall utilities.

4. Using the range of utility values, determine the value of information as discussed in Chapter 4. If value of information indicates that using the current interaction pattern, designer's design objectives are not satisfied and there is a need to increase interaction details, then go to the next interaction pattern and follow steps 2-4.

These four steps are used for determining the right level of simplification of interaction patterns. These steps are discussed in detail in Sections 5.3 and 5.4 for scale decoupling and decision decoupling respectively.

5.3 Scale Decoupling

5.3.1 Method for Scale Decoupling

In this section, we discuss the scenario where a designer wants to make a decision using simulation models at multiple scales. Although the simulation models are coupled with each other, the coupling may not be important from the point of view of decisions under consideration. By considering a completely coupled simulation model, the fidelity of simulation is high but the associated time for executing the model is also high. Decoupling of the simulation models at different scales reduces both the accuracy and the time required to execute the simulation models. Hence, the designer is faced with the following meta-level decision involving tradeoff between the costs of employing accurate simulation vs. accuracy for decision making – *what level of coupling between simulation models needs to be considered*.

Since we are considering the case involving only two models, the meta-level decision refers to selection of pattern P1, P2 or P3 that provides enough accuracy for decision making, while minimizing the computational costs. The decision scenario is shown in Figure 5-7, where a decision is modeled in terms of compromise DSP (cDSP) keywords and one of the three different types of model interaction patterns P1, P2, or P3 can be used to make the decision. In order to help designers make this meta-level decision, we employ the four general steps listed in Section 5.2.

Specifically, the details of steps followed for *scale decoupling* are:

Step 1: *Decision formulation:* The first step in the scale decoupling process is to formulate a decision that needs to be made using the simulation models. We use the compromise DSP construct to model design decisions. Compromise DSP construct captures information about design variables, responses, simulation models used for evaluating responses from design variables, designers' preferences, constraints, and goals. Since there is one decision associated with the determination of values for the design variables and another decision corresponding to the meta-level decision (selection of the appropriate interaction pattern), the designers need to formulate preferences for both decisions.

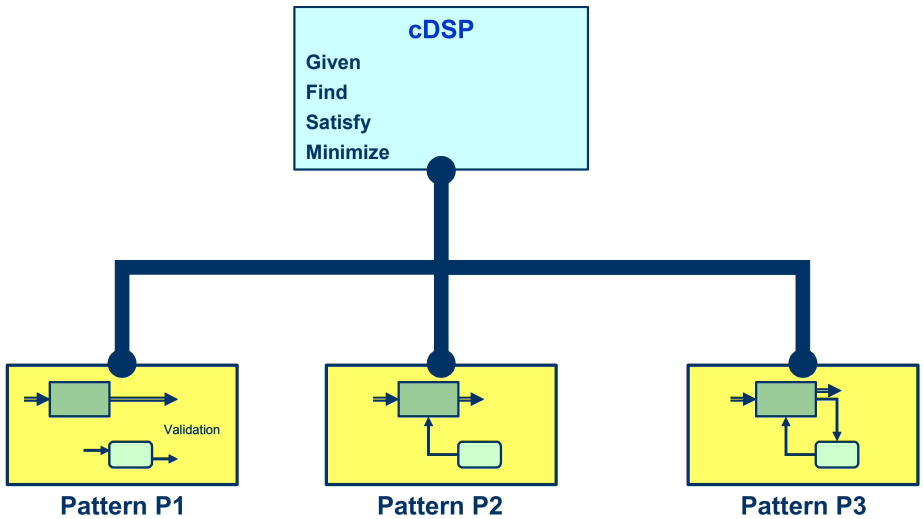


Figure 5-7 - Use of patterns P1, P2, or P3 for making a decision

We model the preferences for achievement of different goals as utility functions. Hence, the decision formulation step involves formulation of utility functions for achievement of performance targets and their deviations from the target. These individual utility functions are combined together to result in an overall utility. The overall utility value indicates the level of fulfillment of designers' objectives. For the meta-level decision, the designers need to determine a cutoff value of the overall utility that if achieved by a simpler interaction pattern would satisfy the designer. This cutoff value of overall utility is used to evaluate the value of information metrics corresponding to different interaction patterns.

<u>Step 2:</u> *Characterization of interaction patterns:* The second step is to characterize the error in predictions from simplified interaction patterns. For example, assuming that pattern P3 is the most accurate model, there is an error introduced due to decoupling of interaction to patterns P2 and P1. Knowledge of this error is used to determine the impact on decision making. The error is quantified as ranges (lower and upper bounds) of values where the response values would lie. The error is generally a function of design variable values. This error is determined either from designers' knowledge, other accurate models, experimental data, or by comparing the predictions with a more accurate model. In this section, we evaluate the error by comparing the model predictions with pattern P3 and then fit a response surface on lower and upper bounds of error as a function of design variables. Since the error due to simplification is the driver for determining the value of information corresponding to interaction patterns, the quality of this error prediction determines the quality of decision making. The estimation of error for each pattern may seem to be an overhead for design but unless the error is quantified, it is not possible to make meta-level decisions in a systematic manner. Further, it is important to note that if this characterization of error is performed once, it can be reused over and over again for designing similar products (re-design) with different specifications that require with making similar decisions numerous times with varying preferences. This reusability is shown in the datacenter example (see Section 5.3.2) for designing datacenters with different preference scenarios. It is shown that different interaction patterns are suitable for different preferences and the knowledge about error due to simplification of interactions is reused in different decisions.

Step 3: *Decision-making using simplified patterns:* After the decision is formulated and the information about error introduced by simplification is known, the designers can select the simplest possible interaction pattern (e.g., pattern P1) and make a decision about design variables. Due to the range of achievable response values, each point in the design space represents a range of overall utility values. Using the range of utility values, a decision is made by selecting the design variable values that maximize the expected value. As discussed before, the robust-decision is formulated by modeling preferences for both mean and variance in the form of utility functions. After making a decision using the simplified pattern, value of information metric is evaluated and the need for considering a more detailed interaction pattern is identified.

Step 4: *Value of Information estimation:* The range of achievable overall utility values is used to evaluate the *achievement* and *opportunity* ratios defined in Chapter 4. These ratios define the value of information generated by a model interaction pattern. Achievement ratio indicates whether designer's objective is fulfilled (based on the cutoff value of overall utility). Opportunity ratio indicates whether there is a hope of achieving higher utility values by changing the design decision. If achievement and opportunity ratios are both high (close to 1), then there is no need to increase the complexity of design process – the current interaction pattern is good enough for making the design decision. If both achievement and opportunity ratios are low (close to 0), then the designer should consider upgrading to an interaction pattern that better represents the physical phenomena and increases the accuracy of response prediction. In that case, the steps 3 and 4 are repeated with the next level of interaction pattern.

The method presented in this section is based on the assumption that information about error due to decoupling is available as bounds on the response. If the error bounds are unavailable, the value of information metric cannot be applied in the manner shown because the metric proposed in this thesis is based on ranges of utility values obtained for different points in the design space. If the decoupling induced imprecision is captured in other mathematical forms such as probability distribution functions, a different value of information metric is required. This defines the scope of application of the proposed method. Further, in this chapter, we do not consider system uncertainty due to noise variables (in addition to the imprecision due to decoupling). Uncertainty due to noise factors can be included in the method in a manner similar to that shown in 9.6, where appropriate level of model refinement is determined in the presence of both uncertainty and imprecision. This extension is left as an opportunity for future work.

5.3.2 Scale Decoupling Example from Datacenter Design

In order to illustrate and validate the steps of scale decoupling listed in Section 5.3.1, we present an example of datacenter cooling system design. Datacenters are computational facilities that consist of huge numbers of data processing units (computers) for high end computing requirements. These facilities range from several square feet to around 5000 square meters. Generally, the computers are stacked vertically in cabinets that are organized horizontally in rows and columns. An example of datacenters is shown in Figure 5-8. Due to the dense packing of these computers in datacenters and their high performance, heat dissipation is a major concern for design of datacenter facilities. Energy costs for cooling the datacenters represent around 40% of their total operation costs. Datacenter designers are concerned with decisions such as number of computers in a cabinet, distance between cabinets, temperature and velocity of cool air to be supplied in a datacenter, etc. The objective is to maximize space utilization, and minimization of costs. Effective design of datacenters is important because of the short lifecycle of the computing equipment. It is estimated that the computing infrastructure changes in about three years on an average.

The thermal behavior of datacenters is dependent on a number of scales that are interlinked with each other (see Figure 5-8). At the room level (~10's of meters), the thermal characteristics of a datacenter depend on the dimensions of the overall facility, arrangement of cabinets in the facility, and the thermal characteristics of individual cabinets. At the individual cabinet level (~1-2 meters), the thermal behavior depends on the number and arrangement of computers in a cabinet, the distance between different computers, the capacity of fans used for drawing air from the cabinets, and the characteristics of each computer. The thermal behavior of each computer (~0.6 meter) is a function of the arrangement of processors and other heat emitting components and the heat generated by each component. The thermal characteristics of the components such as a processor (~35mm) are determined by the component's architecture. Hence, the overall design of the datacenter should be carried out by considering the phenomena at all these scales. In other words, a completely coupled simulation that models phenomena at all these scales would have been the "best" model. However, the disadvantage of such a model is that it would take an extremely long time (at the order of years) to simulate a coupled system. From a meta-design perspective, the designers' needs to decide the appropriate level of complexity of the model that would be good enough for making decisions about the datacenter layout. Not all couplings are important for making design decisions. In order to solve this problem of determining the appropriate couplings between scales that should be considered, we employ the scale decoupling method presented in Section 5.3.1. The first of the four steps detailed in that section is to formulate the decision.

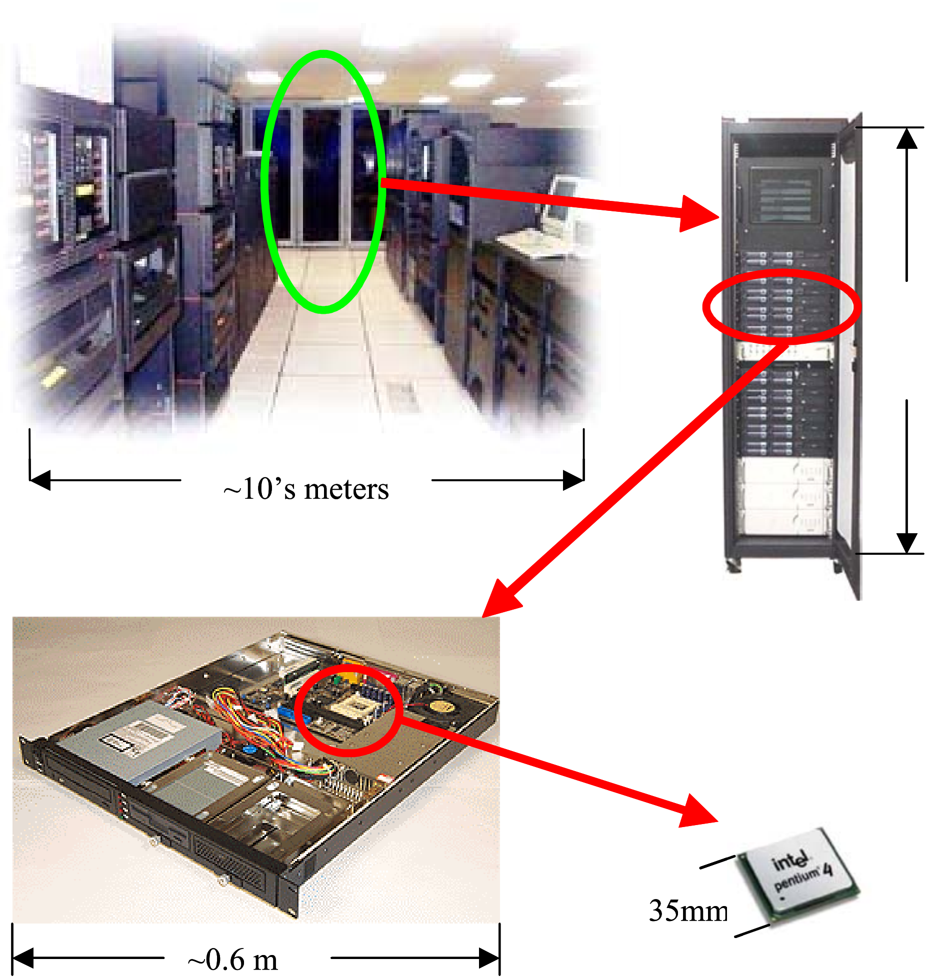


Figure 5-8 – Multiple scales in the datacenter cooling system design

Step 1: Decision Formulation

Consider a design scenario where a designer is interested in designing the air conditioning system for a datacenter. The decision is formulated as a compromise decision support problem. The detailed decision formulation is shown in Table 5-3. The temperature and velocity of air entering the cabinets are chosen as design variables. The objectives include effective cooling of the surface of computers (i.e., minimization of temperature on the surface) and minimization of cooling cost. The preferences for both these objectives defined as utility values and are shown in Figure 5-9. These preferences correspond to risk-averse nature of designers. These utility functions for goals of average temperature and cost are combined together by a taking a weighted average of individual utility values. The assumption that allows for such a combination is that the utility values for individual goals are independent of each other. This condition is also known as preference independence of the goals.

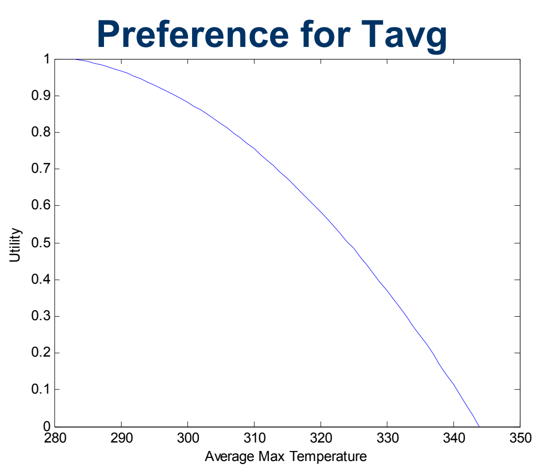


Figure 5-9 - Preferences for average temperature and cost indicator

Table 5-3 – Decision formulation for datacenter design

Decision formulation for datacenter design
Given
Simulation models at both levels
Preferences and targets on average temperature achieved (T) and Cost Indicator (C) as shown in Figure 5-9
$U_{Cost} = \begin{cases} -0.8\left(\frac{C-C_{min}}{C_{max}-C_{min}}\right)^2 - 0.2\left(\frac{C-C_{min}}{C_{max}-C_{min}}\right) + 1 & C_{min}<C<C_{max} \\ 0 & C \geq C_{max} \\ 1 & C \leq C_{min} \end{cases}$
$C_{min}=1, C_{max}=100$
$U_{Temp} = \begin{cases} -0.8\left(\frac{T-T_{min}}{T_{max}-T_{min}}\right)^2 - 0.2\left(\frac{T-T_{min}}{T_{max}-T_{min}}\right) + 1 & T_{min}<T<T_{max} \\ 0 & T \geq T_{max} \\ 1 & T \leq T_{min} \end{cases}$
$T_{min}=283K, \ T_{max}=344K$
Preferences related to imprecision in temperature prediction
$U_{Temp\_uncertain} = \begin{cases} -0.8\left(\frac{T_{Error}-T_{min}}{T_{max}-T_{min}}\right)^2 - 0.2\left(\frac{T_{Error}-T_{min}}{T_{max}-T_{min}}\right) + 1 & T_{min}<T<T_{max} \\ 0 & T \geq T_{max} \\ 1 & T \leq T_{min} \end{cases}$
Find
Values of design variables T_{in}, V_{in}
Values of deviation variables d^-_{Cost}, d^-_{Temp}, $d^-_{Temp\_Uncertain}$
Satisfy
Goals forT and Cost indicator
$U_{Temp} + d^-_{Temp} - d^+_{Temp} = 1$
$U_{Cost} + d^-_{Cost} - d^+_{Cost} = 1$
$U_{Temp\_Uncertain} + d^-_{Temp\_Uncertain} - d^+_{Temp\_Uncertain} = 1$
Bounds on design variables

Decision formulation for datacenter design
$T_{in} = [273 \quad 300]K$ $V_{in} = [1 \quad 2.5] m/\sec$ *Bounds on deviation variables* $d^+_{Temp}, d^-_{Temp}, d^+_{Cost}, d^-_{Cost}, d^+_{Temp\_Uncertain}, d^-_{Temp\_Uncertain} \geq 0$ $d^+_{Temp}.d^-_{Temp} = 0$ $d^+_{Cost}.d^-_{Cost} = 0$ $d^+_{Temp\_Uncertain}.d^-_{Temp\_Uncertain} = 0$ $k_1 + k_2 + k_3 + k_4 + k_5 + k_6 = 1$
Minimize   *Deviation from target* $Z = k_1 d^+_{Temp} + k_2 d^-_{Temp} + k_3 d^+_{Cost} + k_4 d^-_{Cost} + k_5 d^+_{Temp\_Uncertain} + k_6 d^-_{Temp\_Uncertain} \geq 0$

In addition to the mathematical models of preferences, it is assumed that models for thermal behavior are also available. Although computational fluid dynamics models for predicting the thermal and flow behavior of air in a datacenter can be developed for at all the four scales, we consider models at only two scales – *cabinet level* and *computer level*. This method and the results can be easily extended to problems involving more than two scales based on the discussion in Chapter 3. In that chapter, we show how any process network involving more than two components can be successively viewed and analyzed as systems with two-components interacting with each other.

A schematic for the *cabinet level model* is shown in Figure 5-10. In the figure, a cross section with two cabinets (one on left and another on right) is shown. The cabinets are separated by a distance D, which varies between 0.5 and 2 meters. The width of a cabinet is 1 meter and consists of a number of shelves on top of each other. The computers are oriented horizontally and placed on the shelves. The distance between different shelves on a stack is denoted by H. The number of shelves in a cabinet is denoted by N and can be varied. The cooling air enters from the bottom of the floor through perforated tiles.

The velocity and temperature of air entering the room through the floor are denoted as Vin and Tin respectively. It is assumed that the top surface of each shelf generates a fixed and uniform heat flux. It is important to note that this is a 2-dimensional model.

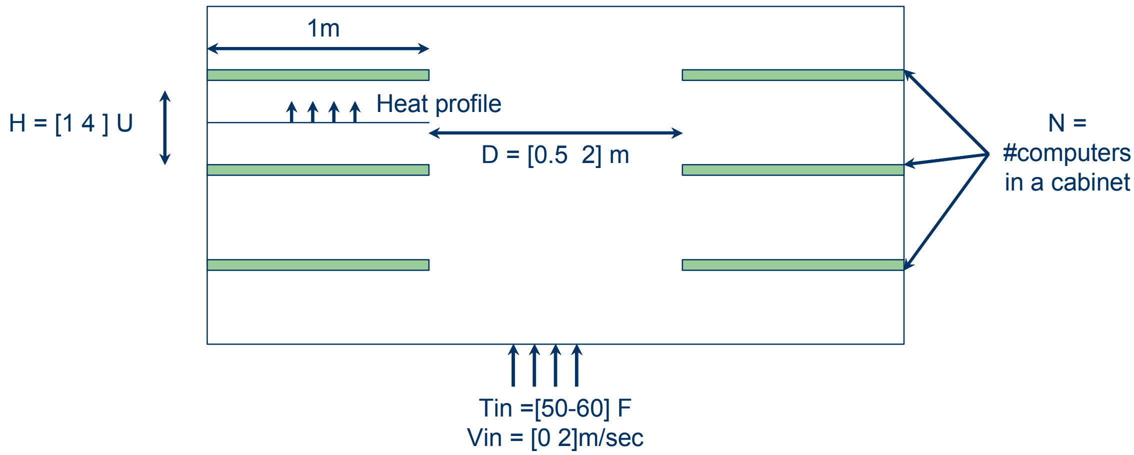


Figure 5-10 - Ilustration of datacenter design variables and parameters – Cabinet Level Analysis

A schematic for the *computer level model* is shown in Figure 5-11. In the figure, a volume of air between two adjacent shelves on the cabinet is modeled. The length, width, and height of the volume are represented as L, W, and H respectively. Four computers approximated by rectangular prisms are placed on the lower shelves. The dimensions of each computer are denoted as l, w, and h. The horizontal distance in x and y directions of computers from the centerline are d1 and d2 respectively. It is assumed that the heat is generated on the top surface of each computer. This assumption is better than the cabinet level model, where the heat generation is assumed to be from the complete shelf surface. The air enters from the left hand side of the air volume and exits from the right. The other two vertical surfaces are assumed to have symmetry condition and the top surface of the volume has a no-slip boundary condition. The air outlet side has a velocity boundary condition because we can control the velocity of air outlet by controlling the fan speed.

Strictly speaking, both these models are coupled with each other because the cabinet level model provides the boundary conditions (inlet air velocity, temperature, etc.) for computer level model, which is important for the accurate prediction of conditions inside the computer level model. The computer level model, in turn, provides a better 3-D description of the heat generation and the pressure drop across a shelf in the cabinet. However, from the meta-level perspective, our objective is to determine which level of coupling is important to consider for the decision.

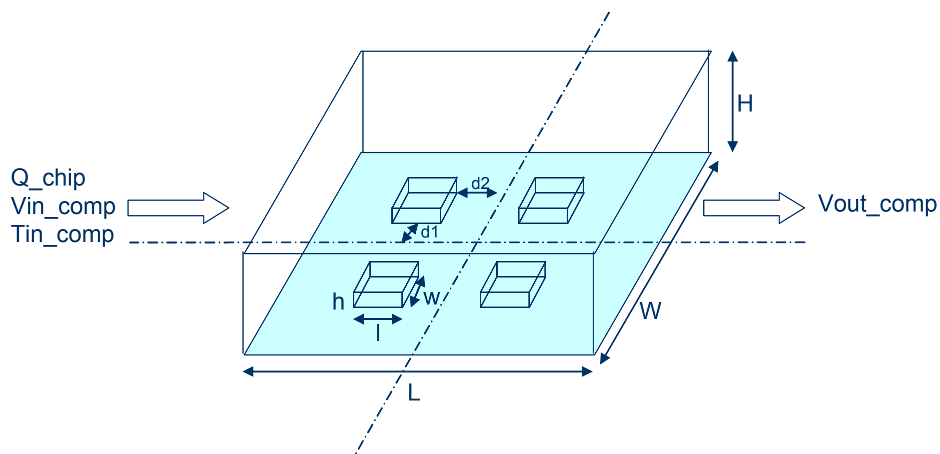


Figure 5-11 - Ilustration of datacenter design variables and parameters – computer level analysis

Step 2: Implementation and Characterization of Interaction Patterns

The next step is to characterize the error in interaction patterns between the two different scales of models. The independent interaction pattern (pattern P1) is shown in Figure 5-12, where the cabinet level model and computer level model are shown. Since this is the simplest interaction pattern P1, there is no information flow between the two models. The cabinet level model has input parameters of distance between cabinets (D), vertical distance between two shelves (H), thickness of each computer (T), inlet temperature (Tin), inlet air velocity (Vin), heat flux generated by the computers (Q), and outlet velocity (Vout). The outputs of the cabinet level simulation are maximum temperature on the surface of computers, and the average temperature on the computer surface.

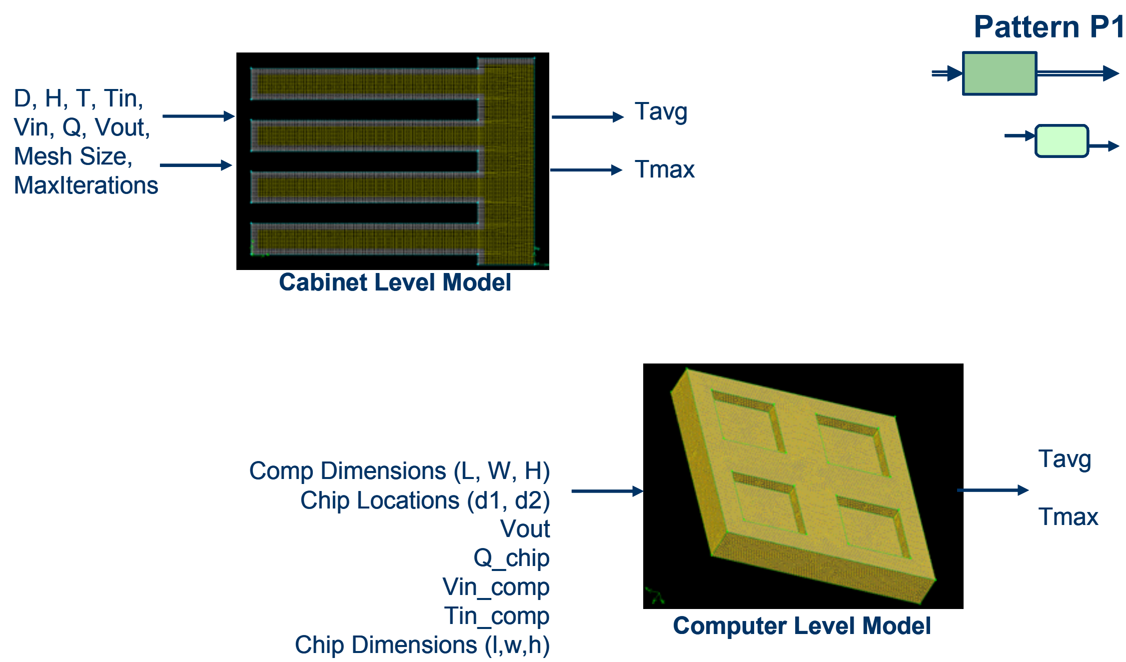


Figure 5-12 – Datacenter analysis representing pattern P1

The computer level simulation has following inputs - computer dimensions (l, w, h), location of computers on the shelf (d1, d2), outlet velocity, heat flux generated by each computer (Q_chip), inlet velocity for each computer (Vin_comp), inlet air temperature for each computer (Tin_comp), and the dimensions of space available on each shelf (L, W, H). Outputs of computer level simulation include the average temperature on the surface of each computer (Tavg), and the corresponding maximum temperature (Tmax). The readers may note that the outputs from both these simulation models are the same. Both these models are created in FLUENT software. Since there is no interaction between the two simulation models, the inlet temperature and velocity in the computer level model is assumed to be the same as conditions at the air inlet into the room. This assumption is necessary for independent pattern because the information is not available for each individual shelf. It is assumed that the boundary conditions at all the shelves are the same.

In the dependent interaction pattern (pattern P2) shown in Figure 5-13, there is one way flow of information between the models. In this case, we model the flow of information from the cabinet level model to the computer level model. The information about the properties (air velocity profile and temperature) at the inlet of each shelf is used as boundary conditions for the computer level model. The outputs from the computer level model are then used for decision making. In the completely coupled pattern (pattern P3 shown in Figure 5-14), the information flows in both directions – from the cabinet level model to computer level model and vice versa. The information about boundary conditions is passed from cabinet level model to computer level model and the information about actual temperature profile for each computer is passed from the computer level model to cabinet level model. One option for developing the completely coupled model is to model the information flow between the two models and then iterating until the output values of temperature converges to a single value. Another option for modeling the coupled phenomenon is to develop a complete CFD model for the cabinet and the computers by including all the details in the same model. Such a model for the datacenter is shown in Figure 5-15. In this chapter, we use this complete model as the coupled model.

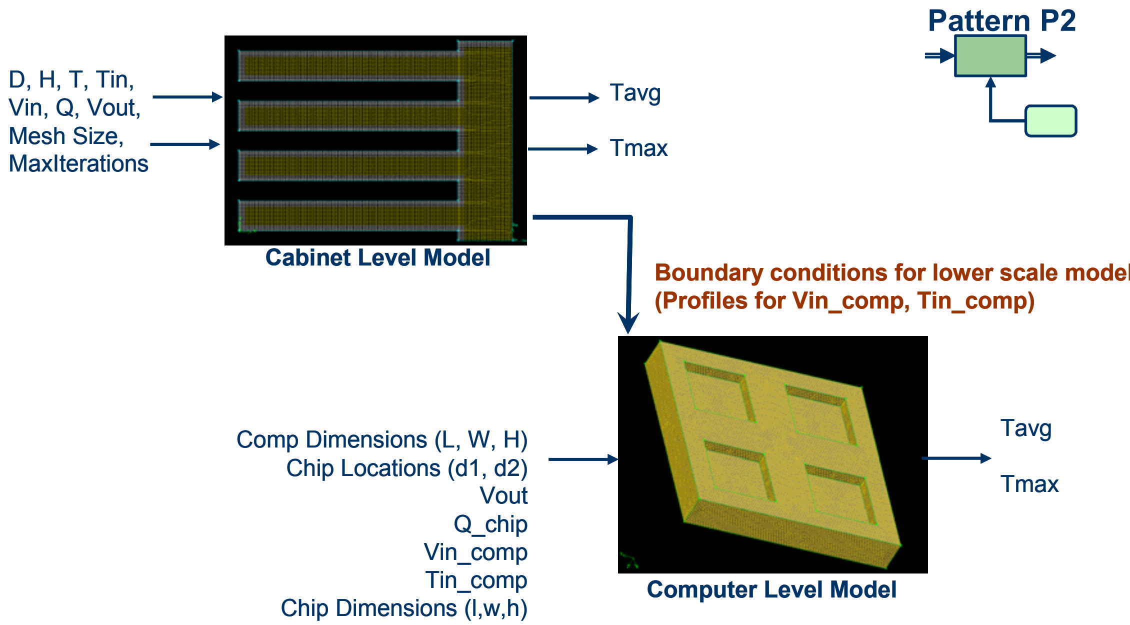


Figure 5-13 – Datacenter analysis representing pattern P2

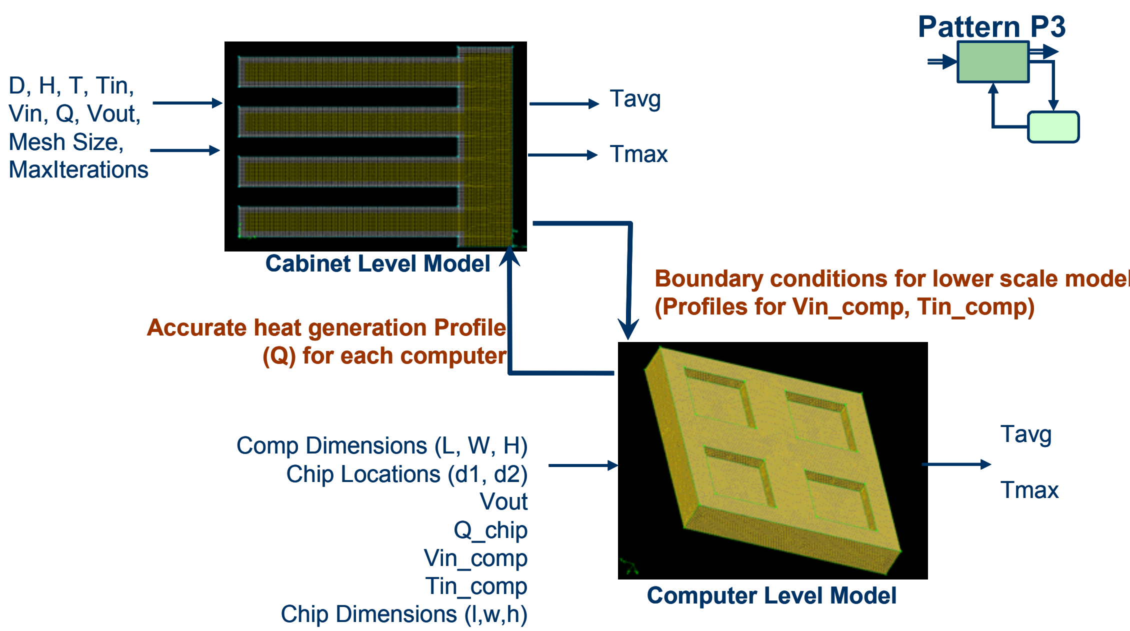


Figure 5-14 - Datacenter analysis representing pattern P3

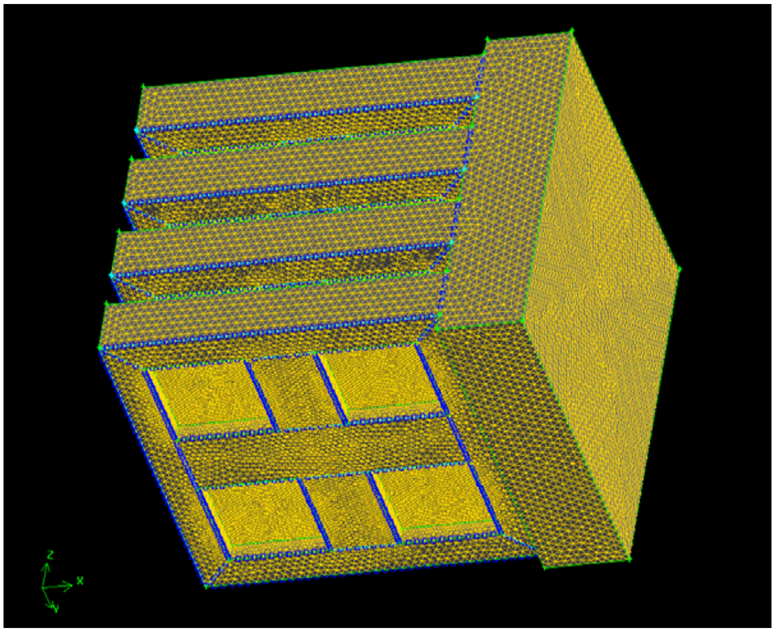


Figure 5-15 – Coupled model representing pattern P3

The mathematical expressions for the response surfaces of average and maximum temperature predicted by simulation models using model interaction patterns P1, P2, and P3 are shown in Table 5-4, Table 5-5, and Table 5-6 respectively.

Table 5-4 – Temperature response surface for pattern P1 and associated error functions

$T^{P1}_{avg} = 299.986 + 10.936\,T_{in} - 3.679\,{}_{Vin} + 0.484\,V^2_{in} + 3.731\,T_{in}{*}V_{in}$
$T^{P1}_{max} = 311.760 + 7.271T_{in} - 6.176V_{in} + 1.189T^2_{in} - 1.926V^2_{in} + 7.176T_{in}V_{in}$
$Error^{P1}_{T_{avg}} = 17.8637 - 8.9684T_{in} + 1.9198V_{in} - 0.0409T^2_{in} - 0.1279V^2_{in} - 2.8316T_{in}V_{in}$
$Error^{P1}_{T_{max}} = 19.7872 - 5.9433T_{in} + 1.0654V_{in} - 0.3336T^2_{in} + 5.0499V^2_{in} - 6.7084T_{in}V_{in}$

Table 5-5 - Temperature response surface for pattern P2 and associated error functions

$T^{P2}_{avg} = 320.042 + 3.571T_{in} - 6.075V_{in} + 0.012T^2_{in} - 5.663V^2_{in} + 6.362T_{in}V_{in}$
$T^{P2}_{max} = 330.710 + 2.558T_{in} - 5.358V_{in} + 1.031T^2_{in} - 3.997V^2_{in} + 4.185T_{in}V_{in}$
$Error^{P2}_{T_{avg}} = -2.192 - 1.602T_{in} + 4.315V_{in} - 0.053T^2_{in} + 6.02V^2_{in} - 5.46238T_{in}V_{in}$
$Error^{P2}_{T_{max}} = 0.837 - 1.230T_{in} + 0.247V_{in} - 0.175T^2_{in} + 7.12V^2_{in} - 3.7181T_{in}V_{in}$

Table 5-6 - Temperature response surface for pattern P3

$T_{avg}^{P3} = 317.850 + 1.968T_{in} - 1.760V_{in} - 0.041T_{in}^2 + 0.356V_{in}^2 + 0.899T_{in}*V_{in}$
$T_{max}^{P3} = 331.547 + 1.328Tin - 5.110V_{in} + 0.856T_{in}^2 + 3.124 * V_{in}^2 + 0.467 * T_{in}V_{in}$

Note that T_{in} and V_{in} are normalized between -1 and 1 corresponding to the range of [273 300]K for T_{in} and [1 2.5]m/sec for V_{in}. In addition to the predicted value, the absolute value of error in predicted value is also listed in the tables. The error associated with pattern P3 is zero because it is assumed to be the perfect model. The temperature predictions for average temperature as a function of inlet temperature and velocity for the three patterns is plotted in Figure 5-16.

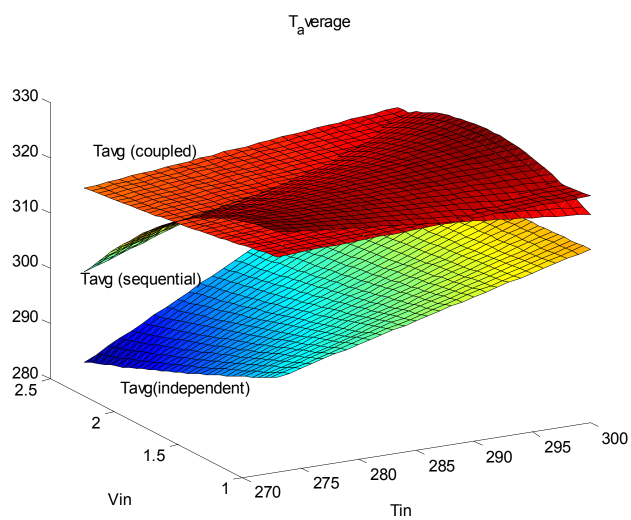


Figure 5-16 – Average temperature predicted using patterns P1 (independent), P2 (sequential), and P3 (coupled) as a function of design variables – inlet temperature and velocity

Results and Validation

The results from decision making using the different interaction patterns – P1, P2, and P3 are shown in Table 5-7. In this table, the outcomes of decisions are shown in terms of design variable values (inlet air temperature – Tin, inlet air velocity - Vin), corresponding response values (average maximum temperature achieved – Tavg, and cost indicator), and the overall utility values corresponding to the response values. Associated with each decision, the value of information is evaluated using the ex-post range, the opportunity ratio (R1) and the achievement ratio (R2) (for value of information metrics, refer Section 4.3). Different decision scenarios are considered. The design scenarios differ in the weights assigned to temperature and cost goals.

Table 5-7 – Results from decision making using patterns P1, P2, and P3 for different preference scenarios

	Pattern	Decision Variables		Response Variables		Overall Utility	Value Metrics		
		Tin	Vin	Tavg	Cost Indicator		Ex-Post Range	R1	R2
w_cost=0.0	P1	273	2.5	282.12	92.5	0.8845	0.30063	1	0.99789
	P2	273	2.5	298.38	92.5	0.80686	0.1993	1	0.99692
	P3	273	2.5	313.54	92.5	0.69938	0	1	0
w_cost=0.1	P1	273	2.1	285.7	77.7	0.81371	0.28046	0.98112	0.81615
	P2	273	2.5	298.38	92.5	0.73935	0.17937	1	0.68013
	P3	275	2	315.34	70	0.64937	0	1	0
w_cost=0.2	P1	273	1.3	293.67	48.1	0.78844	0.2487	0.99889	0.79325
	P2	286	2.5	307.94	60	0.69576	0.11372	0.87316	0.44203
	P3	297	2.05	318.99	26.65	0.66173	0	1	0
w_cost=0.3	P1	273	1	296.94	37	0.78937	0.21451	1	0.86326
	P2	300	1	317.67	10	0.71034	0.043896	1	0.72965
	P3	300	1.8	319.72	18	0.6954	0	1	0
w_cost=0.4	P1	282	1	301.75	28	0.79736	0.16667	0.94876	1
	P2	300	1	317.67	10	0.74818	0.037626	1	1
	P3	300	1.5	320.1	15	0.73171	0	1	1
w_cost=0.5	P1	292	1	307.08	18	0.81732	0.11515	0.90602	1
	P2	300	1	317.67	10	0.78602	0.031355	1	1
	P3	300	1.15	320.69	11.5	0.77037	0	1	1
w_cost=0.6	P1	300	1	311.35	10	0.84625	0.072604	0.86754	1
	P2	300	1	317.67	10	0.82386	0.025084	1	1
	P3	300	1	320.99	10	0.81115	0	1	1
w_cost=0.7	P1	300	1	311.35	10	0.87849	0.050725	1	1
	P2	300	1	317.67	10	0.86169	0.018813	1	1
	P3	300	1	320.99	10	0.85217	0	1	1
w_cost=0.8	P1	300	1	311.35	10	0.91073	0.033817	1	1
	P2	300	1	317.67	10	0.89953	0.012542	1	1
	P3	300	1	320.99	10	0.89318	0	1	1
w_cost=0.9	P1	300	1	311.35	10	0.94297	0.016908	1	1
	P2	300	1	317.67	10	0.93737	0.0062709	1	1
	P3	300	1	320.99	10	0.93419	0	1	1
w_cost=1.0	P1	300	1	311.35	10	0.97521	0	1	1
	P2	300	1	317.67	10	0.97521	0	1	1
	P3	300	1	320.99	10	0.97521	0	1	1

Consider a single decision scenario where the weight for cost goal is 0.2 (which implies that the weight for temperature goal is 0.8). Following the method proposed in Section 5.3.1, we first select the simplest interaction pattern – P1 and make a decision about design variable values. The selected values of inlet air temperature (Tin) and inlet air velocity (Vin) are 273 K and 1.3m/sec respectively. The overall utility predicted by the pattern P1 is 0.78844. For this value of the decision point, the ex-post range is equal to 0.2487. Ex-post range represents the upper bound on value of information that can be achieved by making the interaction pattern more accurate. The opportunity ratio is equal to 0.9988. Since the value is close to 1, it indicates that there is low hope of improving the design solution by selecting another point in the design space. The achievement ratio is equal to 0.79325, which indicates the level of achievement of designer's cutoff value for overall utility.

Since the ex-post range is high, based on implications of these metrics listed in Table 4-3, the confidence in this solution is low. Hence, the designer should add more information and refine the interaction pattern from P1→P2. Using the interaction pattern P2, the design variable values are 286 K, and 2.5 m/sec. The overall utility value predicted is 0.6957 and the ex-post range is 0.1137. If the interaction pattern is refined further (P2→P3) to include complete coupling, the decision is as shown in Table 5-7. The ex-post range is 0.0, which is obvious because the completely coupled interaction pattern is considered the most accurate model and error is calculated based on this pattern.

It is observed from the results in Table 5-7, for preferences corresponding to weight of cost between 0.0 and 0.2, the ex-post range values for both patterns P1 and P2 are greater than 0.1. Hence, the value of adding more information is high in those scenarios. Therefore, pattern P3 is required to make decisions in those preference scenarios. In scenarios where the weight for cost is between 0.3 and 0.6, the ex-post range for pattern P1 is greater than 0.1 but the ex-post range for pattern P2 is less than 0.1. Hence, value of added information from P1→P2 is high but from P2→P3 is low. Therefore in those scenarios, the designers may use pattern P2 for decision making. In the remaining preference scenarios where the weight for cost goal is greater than 0.6, the ex-post value for adding information by moving from P1→P2 is less than 0.1. Hence, pattern P1 is good enough for decision making. The conclusion based on the results in Table 5-7 is that if the weight for cost is between 0.0 and 0.2, pattern P3 is suitable for decision making. Pattern P2 is suitable if the weight for cost is between 0.3 and 0.6, and pattern P1 and suitable if the weight for cost is above 1.0 (given all the other factors in the decision including constraints, preferences for individual goals, bounds on design variables, etc. remain the same). This trend is intuitive because when the weight for cost is low, accurate temperature prediction becomes important. Hence, the designers need to use more accurate interaction patterns. This also indicates that there is a significant dependency of preference on the selection of appropriate model interaction pattern. Hence, if the models are characterized in terms of error once, then this information can be used again and again for different decision making scenarios.

It is important to note that this meta-level decision about the appropriateness of a model interaction pattern is solely based on the value of information metric. It is based on selecting the simplest model first and then making a decision whether there is a need to refine it further. Hence, this method supports meta-design without executing all the available design process options. In order to gain a deeper understanding of the decision making behavior, the design variables, responses and individual utility functions for temperature and cost goals are plotted in Figure 5-17. Decisions made by using different interaction patterns are shown for different weights for the goals.

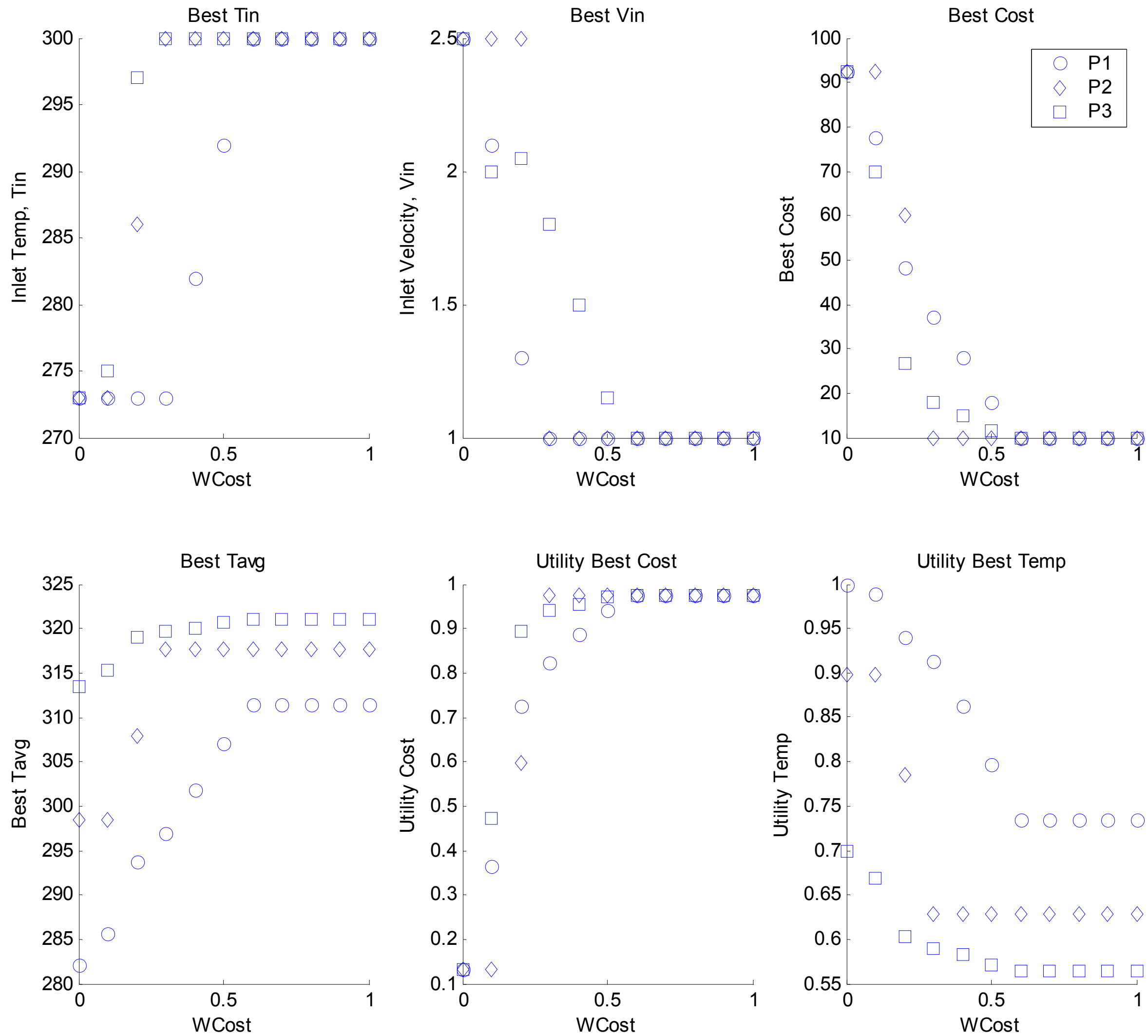


Figure 5-17 – Results from scale decoupling

It is important to note that when the weight for cost goal is low (between 0.0 and 0.2), the values of design variables (Tin, Vin) predicted by any of the three interaction patterns is the same. Hence, no matter which interaction pattern is chosen for making the decision, the actual performance of the system would be the best possible. In other words, even if the decision is made by pattern P1, the system will behave in a way as if the decision is made by pattern P3. Hence the *actual* value of information as shown in Figure 4-4 is zero. However, the fact is that designer does not know how the system behaves as predicted from the most accurate model. All he/she is aware of is the information generated by the interaction pattern (and the associated error bounds if the models are characterized for error). This means that the uncertainty is high. This uncertainty is captured by the ex-post value of information metric. The designers can use this upper bound on value of information for making meta-level decisions. A similar trend is apparent for decision scenarios where the weight of cost greater than 0.6. The design variable values selected by patterns P1, P2, and P3 are the same, implying zero value of information. This is reflected in the low values of ex-post utility values for these decision scenarios. The design variable values selected by different interaction patterns are different for the decision scenarios where weight for cost lies between 0.2 and 0.6.

Two different phenomena result in prediction of the same values of design variables at the two extremes of weight for cost goal. At the lower values of weight for cost goal, the temperature goal dominates the decision. Although error is introduced due to simplification of interaction pattern the trend remains the same – average maximum temperature is lowered by lowering the inlet temperature and increasing the inlet velocity. Hence, the chosen decision point is the lower bound of inlet air temperature and upper bound of inlet air velocity. In other words, although the error is high, the trend in each interaction pattern is the same, resulting in the same decision. When the weight for cost goal is high, the error due to simplification of interaction pattern is low because the cost model in all the three interaction patterns is the same. Hence, the decision made by different interaction patterns is the same. This also indicates the importance of the value of information metric. The example shows that development of perfect model is not required for decision making. Error is not the only criterion for selecting model for making a decision.

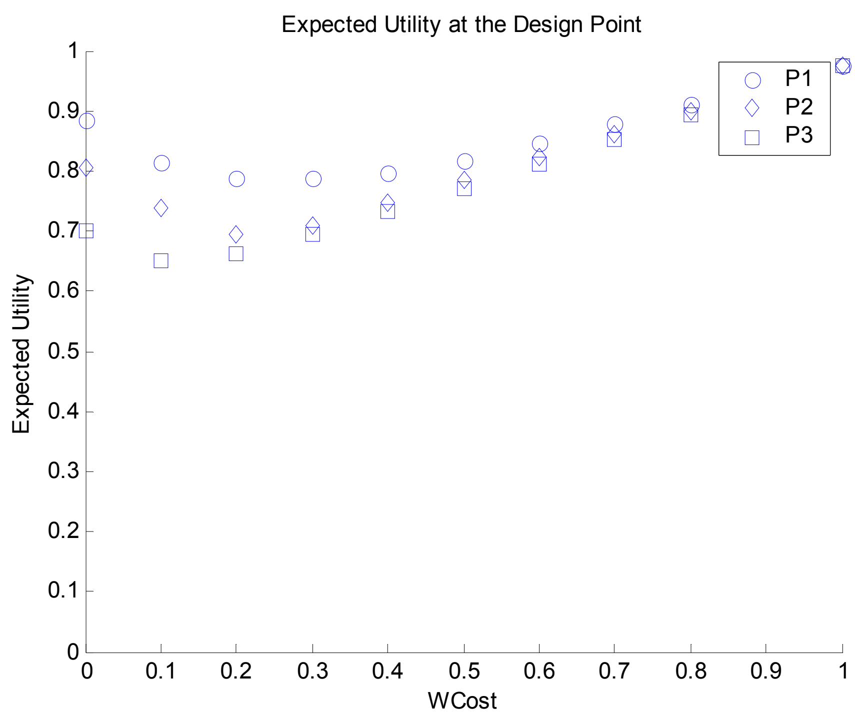


Figure 5-18 – Overall expected utility with increasing weight to cost objective

The opportunity ratio (R1) indicates whether there is a possibility of existence of points in the design space with better overall utility. It may increase or decrease with use of better interaction patterns. It helps designers in determining regions where designers should invest their efforts and get more information about the design space. If the opportunity ratio is low, only then the designers should refine simulation models. If the opportunity ratio is close to 1, then the improvement of simulation models is not important. The opportunity ratio in this case is close to 1 in most of the preference scenarios. The opportunity ratio is equal to 1 for all the preference cases using pattern P3, indicating that no improvement is achievable. The achievement and opportunity ratios for different weights for the cost objective are shown in Figure 5-20 and Figure 5-21 respectively. Note that the ex-post value for pattern P3 is always 1, which is expected because the error in the models is calculated based on this pattern. The ex-post value for sequential pattern P2 is slightly higher than the pattern P3 and that of pattern P1 is higher than P2, indicating that improvement in decision while going from pattern P1 → P3 is more than the improvement in decision while going from P2→ P3.

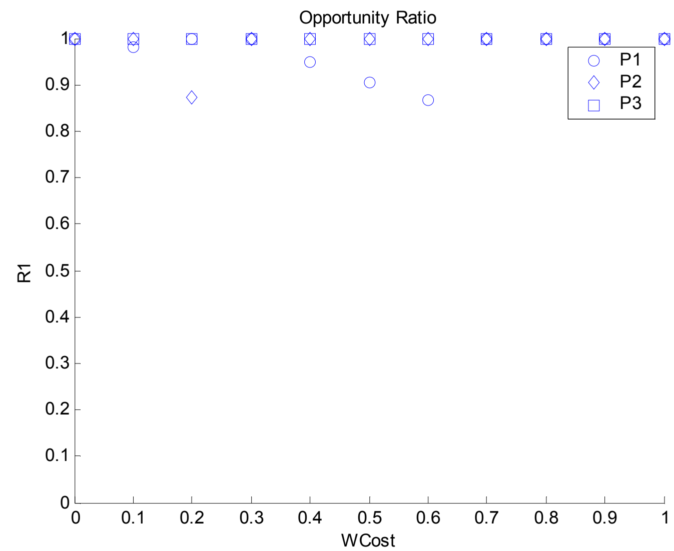


Figure 5-19 – Opportunity Ratio with increasing weight to cost objective

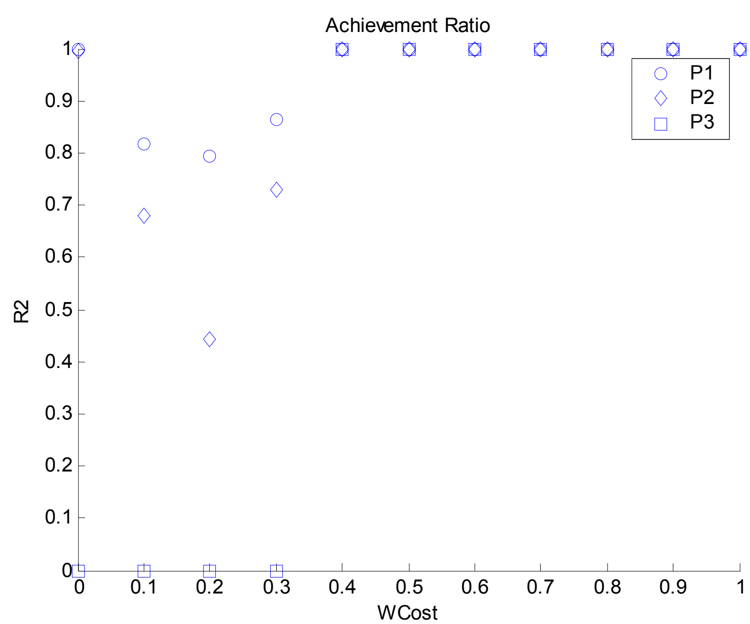


Figure 5-20 – Achievement Ratio with increasing weight to cost objective

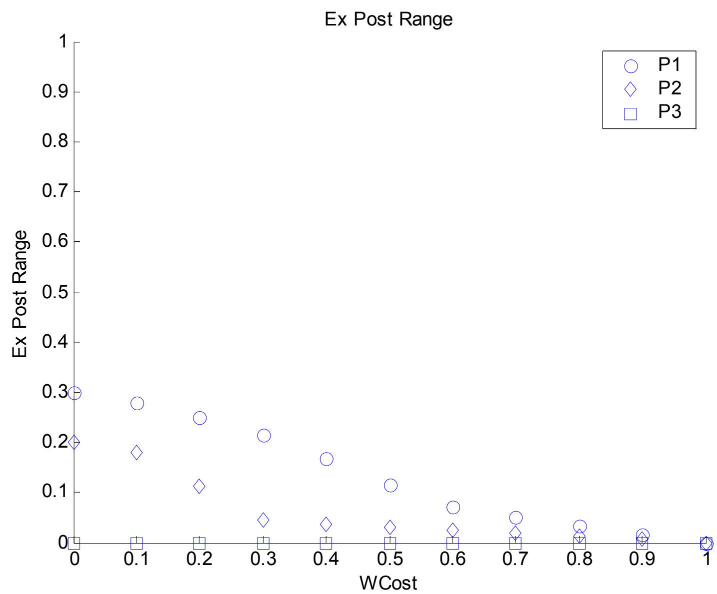


Figure 5-21 – Ex- Post Range with increasing weight to cost objective

5.4 Decision Decoupling

5.4.1 Method for Decision Decoupling

In this section, we discuss a scenario where a designer (or multiple designers) is interested in making multiple decisions about the product. Each of these decisions involves selecting the values of design variables while satisfying the design objectives in the best possible manner. These decisions may be coupled with each other due to the coupling between the physical behavior of the system, or due to the dependencies between preferences in two decisions, or due to common set of constraints. This coupling requires multiple decisions to be considered simultaneously and increases the complexity in design process. Although the decisions are coupled with each other, the effect of this coupling may be low on the designers' decision making capability. The question that arises is – *What level of **coupling between decisions** should be preserved between multiple decisions in order to simplify the design process without affecting on design* *decision?* It is this question that we address in this section. The question is answered in the context of interaction patterns discussed in Section 5.2.4. The three interaction patterns considered in this section are independent decisions (pattern P4), sequential decisions (pattern P5), and coupled decisions (pattern P6). It is important to note that in this section, we only consider the scenario with interactions between two decisions. If there are more than two decisions in the design process, the process can be hierarchically viewed as two decisions, where each decision further consists of multiple decisions. This generalization of the proposed approach using interaction patterns is discussed in Chapter 3. The decision interaction patterns are shown in Figure 5-22. The three patterns show different types of interactions between two decisions. It is emphasized here that there are two different types of decisions to be made – product decisions (determining the value of design variables that define the product form), and meta-level decision (determining which interaction pattern to choose between the product decisions). Although selecting an appropriate model interaction pattern for making a product decision is also a part of the overall problem, for this section, we assume that the decision about the model interaction pattern has already been made using the method discussed in Section 5.3.1. At the end of Section 5.4, however, we discuss how both these meta-level decisions can be made together.

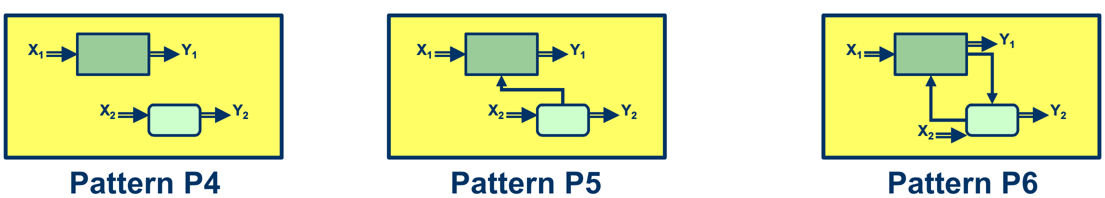


Figure 5-22 – Interaction patterns between decisions

The decision decoupling method is based on the general steps outlined in Section 5.2. Specifically, the details of the steps followed for decision decoupling are discussed next. The steps in the method are in general similar to the steps followed in scale decoupling but there are some differences in terms of the manner in which decisions are characterized.

Step 1: *Formulation of decisions*: The first step in decision decoupling is to formulate the decisions to be made, and the identification of the information flow between them. In this research, the decisions are formulated using the compromise DSP construct. Formulation of the decisions involves identifying the design variables, responses, constraints, simulation-models to be used for predicting the responses, goals, and mathematical formulation of preferences. After the decisions are formulated, the next step is to characterize the interaction patterns in terms of the effect of simplification on the overall utilities of decisions.

Step 2: *Characterization of interaction patterns*: Since the output of each decision is a selected set of values for the design variables considered in that decision, the information flow between decisions consists of values of design variables. Removal of a link between two decisions is equivalent to making the values of design variable flowing from one decision to another imprecise. For example, the output of *Decision A* is a design variable value y that is uses as input to *Decision B* as shown in Figure 5-23(a). If this information link between decision A and decision B is removed (shown in Figure 5-23(b)), then the value of variable y becomes imprecise for decision B. This imprecision is represented as a range of values that the variable y can possibly take. This general idea is used to model the imprecision due to simplification of interactions between decisions. The characterization of imprecision in decision decoupling is simpler than scale decoupling because only the information about bounds on the design variables is required to model imprecision in different interaction patterns.

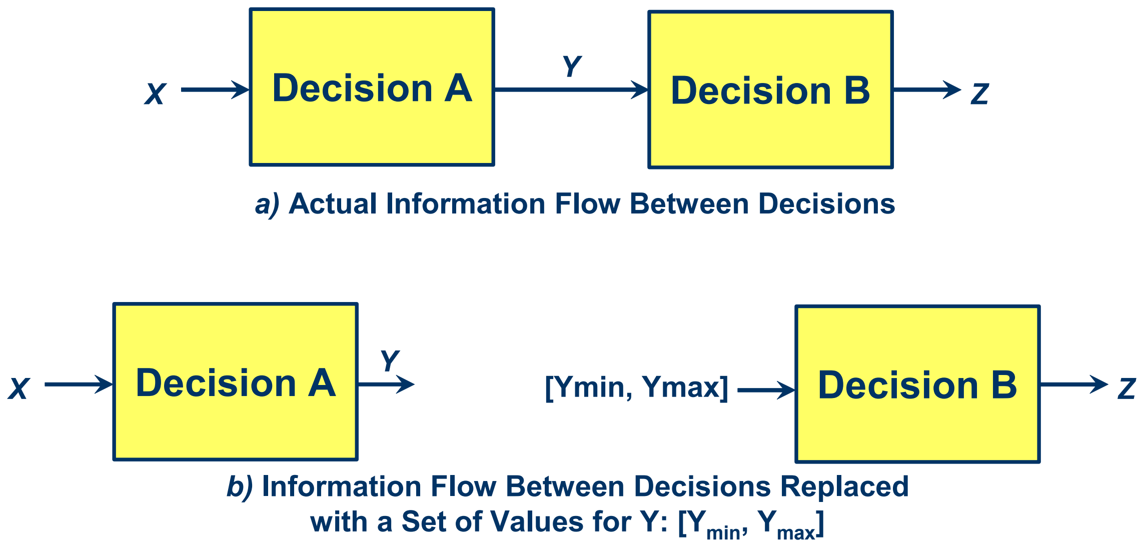


Figure 5-23 - Simplification of Interaction represented as a set of values

Step 3: *Decision making using simplified patterns*: Similar to the scale decoupling, we start with the simplest decision interaction pattern and make decisions such that the expected utility in the presence of imprecision in design variables due to simplification is maximized. The decision can also be made by either *a)* maximizing the average value of the utility obtained in the worst case scenario and that obtained by best case scenario or *b)* a combination of the average and the variation in utility. The consideration of both mean and variation of performance due to variation in design variable values is similar to the Type II Robust design discussed in Section 2.3.

Step 4: *Value of information estimation*: After making the product decisions using simplified interaction patterns, the value of information metrics – ex-post range, opportunity ratio, and achievement ratio are evaluated. These metrics for value of information are then used to make the meta-level decision on whether the simple interaction pattern is good enough for making product decisions or not. For example, if the ex-post value is high, it means that the upper bound on value of information is high. Hence, there is a high chance that the additional information will have a great impact on the decision making capability. If the designer's meta-design decision is to add more information via consideration of interactions between decisions, then the designers select another pattern (P4→P5, P5→P6) and continue the repeat steps 3 and 4.

The application of this method is based on the assumption that the range of design variables is known. This information about design variables is used to replace an information flow between decisions with a set of values that the design variables can take. The decisions are made considering this imprecision in design variables. This implies that if there is no information available about the range of a design variable, then the method cannot be applied. It is also reiterated that in this section, we consider only the imprecision because of not considering the information flows between decisions. The uncertainty inherent to the models is not addressed in this section. This method for decision decoupling is applied to the datacenter design scenario in the following Section 5.4.2.

5.4.2 Decision Decoupling Example for Datacenter Design

In this section, we consider two decisions associated with the design of datacenter cooling system – *a)* decision of inlet air velocity and temperature (cabinet level decision), and *b)* decision of outlet velocity from each computer (computer level decision). All these three design variables can be controlled independently. Air inlet temperature and velocity can be controlled by changing the cooling setting in the room air conditioner. The outlet velocity is controlled by the type and rpm of fans installed on each computer.

The two decisions under consideration are related to design of two separate subsystems – air conditioning and fans on the computer respectively. Both these decisions are dependent on each other because the selection of fan for each computer depends on the air flow conditions around the computers, thereby requiring prior knowledge of the temperature and velocity with which air is sent into the cabinets. The decision about conditions of air inlet depends on the velocity of air exiting the computers and the back pressure created.

The coupling between decisions is caused due to the coupling between physics that govern the subsystem performance of both the air conditioning system and the air flow over the computers. These decisions are also linked because the designers' preferences are defined in terms of average maximum temperature, which is affected by performance of both subsystems. Hence, ideally, the decisions should be represented using the coupled interaction pattern. In this section, we utilize the method for decision decoupling for identifying whether the decisions should be treated in a coupled fashion (pattern P6) or can be simplified into independent or sequential decisions (patterns P4 and P5 respectively).

Table 5-8 – Decision of inlet air velocity and temperature (given that the range of outlet velocity is known)

Decision of inlet air velocity and temperature
Given Simulation models at both levels **Range of outlet air velocities (V_{out})** Preferences and targets on average temperature achieved (T) and Cost Indicator (C) as shown in Figure 5-9 $U_{Cost} = \begin{cases} -0.8\left(\frac{C-C_{min}}{C_{max}-C_{min}}\right)^2 - 0.2\left(\frac{C-C_{min}}{C_{max}-C_{min}}\right) + 1 & C_{min}<C<C_{max} \\ 0 & C \geq C_{max} \\ 1 & C \leq C_{min} \end{cases}$ $C_{min}=1,\ C_{max}=100$ $U_{Temp} = \begin{cases} -0.8\left(\frac{T-T_{min}}{T_{max}-T_{min}}\right)^2 - 0.2\left(\frac{T-T_{min}}{T_{max}-T_{min}}\right) + 1 & T_{min}<T<T_{max} \\ 0 & T \geq T_{max} \\ 1 & T \leq T_{min} \end{cases}$ $T_{min}=283K,\ \ T_{max}=344K$ *Preferences related to imprecision in temperature prediction* $U_{Temp\_uncertain} = \begin{cases} -0.8\left(\frac{T_{Error}-T_{min}}{T_{max}-T_{min}}\right)^2 - 0.2\left(\frac{T_{Error}-T_{min}}{T_{max}-T_{min}}\right) + 1 & T_{min}<T<T_{max} \\ 0 & T \geq T_{max} \\ 1 & T \leq T_{min} \end{cases}$ ***The overall utility function*, which is an average value over all values achieved by varying V_{out}** $U_{overall} = \underset{V_{out}}{avg}(k_1 U_{Cost} + k_2 U_{Temp} + k_3 U_{Temp\_Uncertain})$ $k_1 + k_2 + k_3 = 1$ **Find** *Values of design variables* T_{in}, V_{in} *Values of deviation variables* $d^+_{Overall}$, $d^-_{Overall}$

Decision of inlet air velocity and temperature
Satisfy
Goals forT and Cost indicator
$U_{Overall} + d^{-}_{Overall} - d^{+}_{Overall} = 1$
Bounds on design variables
$T_{in} = [273 \quad 300] K$
$V_{in} = [1 \quad 2.5] m/\sec$
Bounds on deviation variables
$d^{+}_{Overall}, d^{-}_{Overall} \geq 0$
$d^{+}_{Overall} d^{-}_{Overall} = 0$
Minimize
Deviation from target
$Z = d^{+}_{Overall} + d^{-}_{Overall}$

The two decisions formulated using the compromise DSP construct are shown in Table 5-8 and Table 5-9. The interaction patterns - P4, P5, and P6 between these two decisions with corresponding inputs and outputs are shown in Figure 5-24, Figure 5-25 and Figure 5-26 respectively. In the independent interaction pattern P4, the inputs for cabinet level decision are preferences, goals, and a range of values for outlet velocity. Using this range of outlet velocities, the ranges of air inlet temperatures and velocity are determined. The inputs for computer level decision include preferences, goals, and a set of values for inlet temperature and velocity. Note that this range of air inlet conditions is based on the lower and upper bounds on these design variables and this range is independent of the cabinet level decision. Using this range for input air conditions, a range of output velocities is decided upon.

Table 5-9 – Decision of outlet air velocity (given that ranges of other two design variables are known)

Decision of outlet air velocity
Given Simulation models at both levels **<u>Ranges of inlet air velocity (V_{in}) and temperature (T_{in})</u>** Preferences and targets on average temperature achieved (T) and Cost Indicator (C) as shown in Figure 5-9 $$U_{Cost}=\begin{cases}-0.8\left(\frac{C-C_{min}}{C_{max}-C_{min}}\right)^2-0.2\left(\frac{C-C_{min}}{C_{max}-C_{min}}\right)+1 & C_{min}<C<C_{max}\\ 0 & C\geq C_{max}\\ 1 & C\leq C_{min}\end{cases}$$ $C_{min}=1,\ C_{max}=100$ $$U_{Temp}=\begin{cases}-0.8\left(\frac{T-T_{min}}{T_{max}-T_{min}}\right)^2-0.2\left(\frac{T-T_{min}}{T_{max}-T_{min}}\right)+1 & T_{min}<T<T_{max}\\ 0 & T\geq T_{max}\\ 1 & T\leq T_{min}\end{cases}$$ $T_{min}=292K,\ \ T_{max}=344K$ *Preferences related to imprecision in temperature prediction* $$U_{Temp\_uncertain}=\begin{cases}-0.8\left(\frac{T_{Error}-T_{min}}{T_{max}-T_{min}}\right)^2-0.2\left(\frac{T_{Error}-T_{min}}{T_{max}-T_{min}}\right)+1 & T_{min}<T<T_{max}\\ 0 & T\geq T_{max}\\ 1 & T\leq T_{min}\end{cases}$$ ***The overall utility function*, which is an average value over all values achieved by varying T_{in}, V_{in}** $U_{overall}=\underset{T_{in},V_{in}}{avg}(k_1U_{Cost}+k_2U_{Temp}+k_3U_{Temp\_Uncertain})$ $k_1+k_2+k_3=1$ **Find** *Values of design variable* V_{out} *Values of deviation variables* $d^+_{Overall}$, $d^-_{Overall}$

Decision of outlet air velocity

Satisfy

Goals forT and Cost indicator

$$U_{Overall} + d^-_{Overall} - d^+_{Overall} = 1$$

Bounds on design variables

$$V_{out} = [-0.1 \quad -1] m / \sec$$

Bounds on deviation variables

$$d^+_{Overall}, d^-_{Overall} \geq 0$$

$$d^+_{Overall} d^-_{Overall} = 0$$

Minimize

Deviation from target

$$Z = d^+_{Overall} + d^-_{Overall}$$

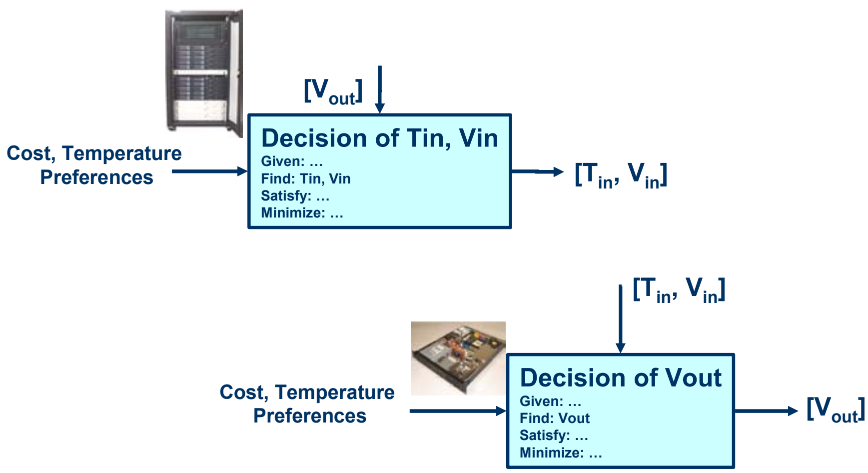


Figure 5-24 - Decision interaction pattern P4

In the sequential interaction pattern (shown in Figure 5-25), it is assumed that the cabinet level decision is made before the computer level decision. In this interaction pattern, the cabinet level decision remains the same as in independent interaction pattern. The only difference here is that instead of taking the complete range of values (lower and upper bounds defined by the design space definition), the output of cabinet level decision (ranges for inlet temperature and velocity) is used as an input to the computer level decision. The information about design variables flows between both decisions in the coupled interaction pattern (see Figure 5-26). The output range of design variables from one decision is an input range for another decision. These coupled decisions can be executed in a number of ways – by combining the decisions into a single decision and executing it as a single decision, by making the two decisions in a sequential manner and iterating the sequence until the range of values converge to a point, or by using game theory based protocols for making coupled decisions. In this section, we use the method where decisions are combined and solved as a single decision.

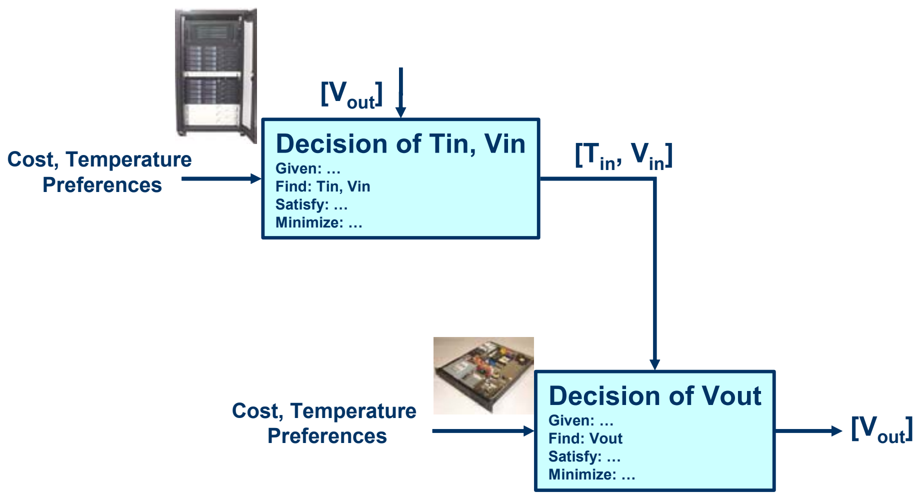


Figure 5-25 - Decision interaction pattern P5

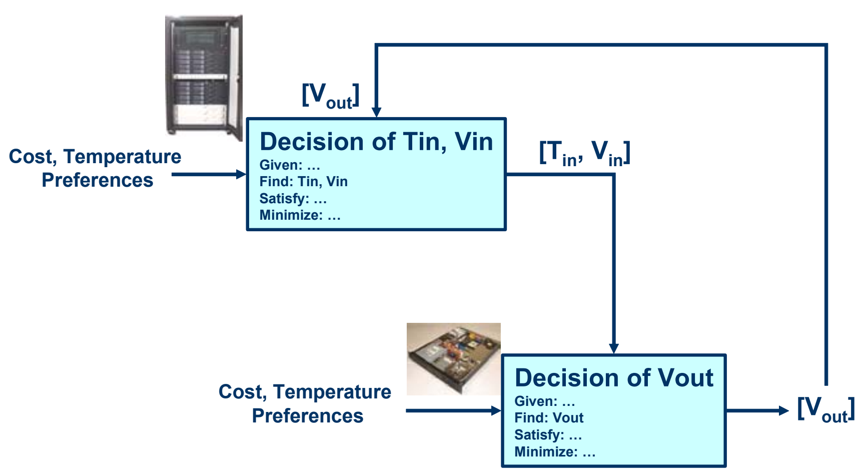


Figure 5-26 - Decision interaction pattern P6

In this section, we assume that the decision about which model interaction pattern to use has already been made. Under this assumption, the task is to *determine the suitable decision interaction pattern for a given model interaction pattern*. The results from decision decoupling are presented in the following. Note that the decisions can be made by using different kinds of model interaction patterns (P1, P2, or P3). Hence, in addition to the consideration of coupling between decisions, we also need to consider the coupling between models that are used to for making individual decisions. In order to decouple these two aspects – decoupling of models and decoupling of decisions, the strategy adopted in this section is to perform decision decoupling for each model interaction pattern separately. The results for decision decoupling using model interaction pattern P1 are shown in Table 5-10, Figure 5-27, and Figure 5-29. The results using sequential model interaction pattern P2 are shown in Table 5-11, Figure 5-28, and Figure 5-30. Finally, the results using coupled model interaction pattern P3 are shown in Table 5-12, Figure 5-31 and Figure 5-32. In the tables Table 5-10, Table 5-11, and Table 5-12, final values of decision variables, the corresponding values of response variables, overall utility and the value of information metrics are shown. These decision values are shown for different values of preferences for cost and temperature goals. Different rows in the table are marked with 'w_cost' values that represent the weight given to the cost goal. The sum of weights for cost and temperature goals is equal to 1. Hence the preference for cost goal increases from the top to bottom of the table. This is to study the impact of changing preferences on the appropriateness of different decision interaction patterns. Figure 5-27, Figure 5-28, and Figure 5-31 are plots of the upper bound of ex-post value of information as the weight for cost goal is increased. Figure 5-29, Figure 5-30, and

Figure 5-32 are plots of design variables, response, and the overall utility as a function of the weight for the cost goal.

Results using Model Interaction Pattern P1

From the results using model interaction pattern P1, it is observed from the results that the ex-post value of information decreases monotonically from pattern P4 to P5 to P6. The ex-post value for pattern P4 is significantly higher as compared to patterns P5 and P6 which implies that the independent decision pattern is not appropriate for decisions under consideration. The ex-post values for decision interaction patterns P5 (sequential) and P6 (coupled) are close to each other and approach zero. Hence, the possibility of improvement in designer's decision from pattern P5 to pattern P6 is little.

Table 5-10 - Comparing the decision interaction patterns using P1 model interaction pattern

	Using Model Interaction Pattern P1									
		Decision Variables			Response Variables		Overall Utility	Value Metrics		
	Pattern	Tin	Vin	Vout	Tavg	Cost Indicator		Ex-Post Range	R1	R2
w_cost=0.0	P4	273	2.5	-0.76	319.68036	92.5	0.87533	0.306263	0.999631	0.97702
	P5	273	2.5	-0.79	313.07376	92.5	0.999227	0	1	1
	P6	273	2.5	-0.8	313.109124	92.5	0.999227	0	1	1
w_cost=0.1	P4	273	1.26	-0.76	319.68036	46.62	0.864008	0.223218	0.959686	1
	P5	273	1.26	-0.55	315.62072	46.62	0.945251	0.019385	0.974106	1
	P6	273	1.2	-0.56	317.896169	44.4	0.944381	0	1	1
w_cost=0.2	P4	273	1	-0.76	319.68036	37	0.852685	0.180673	0.932368	1
	P5	273	1	-0.49	319.588827	37	0.930086	0	1	1
	P6	273	1	-0.52	319.457496	37	0.930063	0	1	1
w_cost=0.3	P4	273	1	-0.76	319.68036	37	0.841363	0.177553	0.939048	1
	P5	273	1	-0.49	319.588827	37	0.916511	0	1	1
	P6	273	1	-0.52	319.457496	37	0.916491	0	1	1
w_cost=0.4	P4	273	1	-0.76	319.68036	37	0.830041	0.250719	0.946141	0.806386
	P5	273	1	-0.49	319.588827	37	0.902936	0	1	1
	P6	273	1	-0.52	319.457496	37	0.902919	0	1	1
w_cost=0.5	P4	276.5	1	-0.76	319.68036	33.5	0.818719	0.32603	0.958947	0.583572
	P5	276.5	1	-0.51	320.518041	33.5	0.893895	0.011789	0.999086	1
	P6	277.5	1	-0.52	320.525415	32.5	0.891487	0	1	1
w_cost=0.6	P4	283.5	1	-0.76	319.68036	26.5	0.807396	0.410111	0.978324	0.458144
	P5	283.5	1	-0.57	321.576811	26.5	0.892249	0.013926	0.997804	1
	P6	284	1	-0.6	321.288043	26	0.888841	0	1	1
w_cost=0.7	P4	291	1	-0.76	319.68036	19	0.796074	0.505205	0.99325	0.389512
	P5	291	1	-0.63	322.611963	19	0.900312	0.016201	0.997543	1
	P6	291.5	1	-0.64	322.402523	18.5	0.897192	0	1	1
w_cost=0.8	P4	299.5	1	-0.76	319.68036	10.5	0.784752	0.613603	0.99926	0.356571
	P5	299.5	1	-0.7	323.189	10.5	0.922025	0.009083	0.997554	1
	P6	300	1	-0.72	323.036843	10	0.918857	0	1	1
w_cost=0.9	P4	300	1	-0.76	319.68036	10	0.77343	0.728519	0.999714	0.339043
	P5	300	1	-0.71	323.077034	10	0.947032	0	1	1
	P6	300	1	-0.72	323.036843	10	0.947032	0	1	1
w_cost=1.0	P4	300	1	-1	319.601	10	0.762107	0.843434	1	0.326293
	P5	300	1	-1	320.749	10	0.975207	0	1	1
	P6	300	1	-1	320.749	10	0.975207	0	1	1

Another observation from the results presented in Figure 5-29 is that the independent pattern P4 provides a decision about Vin, Tin that is close to its coupled counterpart. This implies that the decisions that are made robust to the value of Vout are close to those where the value of Vout is known precisely. This is an important indicator that the decision of Vout can be decoupled from the decision about Vin and Tin. However, the decision of Vout made independently by consideration of robustness to values of Tin and Vin is significantly different from the decision that would have been made in a coupled fashion. This implies that the decision of Vout is hightly dependent on the values of Tin and Vin. This dependence suggests that decisions about Tin and Vin can be made independently but the decision about Vout should be made with the knowledge of Tin and Vin, which imposes a sequential precedence relationship between the two decisions. Hence, the sequential decision pattern P5 should be good enough in this scenario. The appropriateness of sequential decision pattern is also apparent from the closeness of results from sequential and coupled decision patterns. As discussed, the trend is also observed in the values of ex-post values. In the case where a sequential model interaction pattern is chosen, high ex-post values indicate that independent decision pattern P4 is not appropriate. Pattern P5 is appropriate when the weight for cost goal is either greater than 0.8 or less than 0.1. Pattern P6 is appropriate when the weight for cost goal is between 0.1 and 0.8. This is mainly because at lower cost values (0.1 to 0.4), the decision made using patterns P4 and P5 deviate from that using P6 whereas at higher values (0.4 to 0.8), the decisions of design variable Tin made using P4 and P5 deviate from P6. In the case where model interaction pattern P3 is chosen (see Figure 5-31), the decision pattern P5 is good enough only when the weight for cost goal is greater than 0.4. For values less than 0.4, coupled pattern P6 should be used.

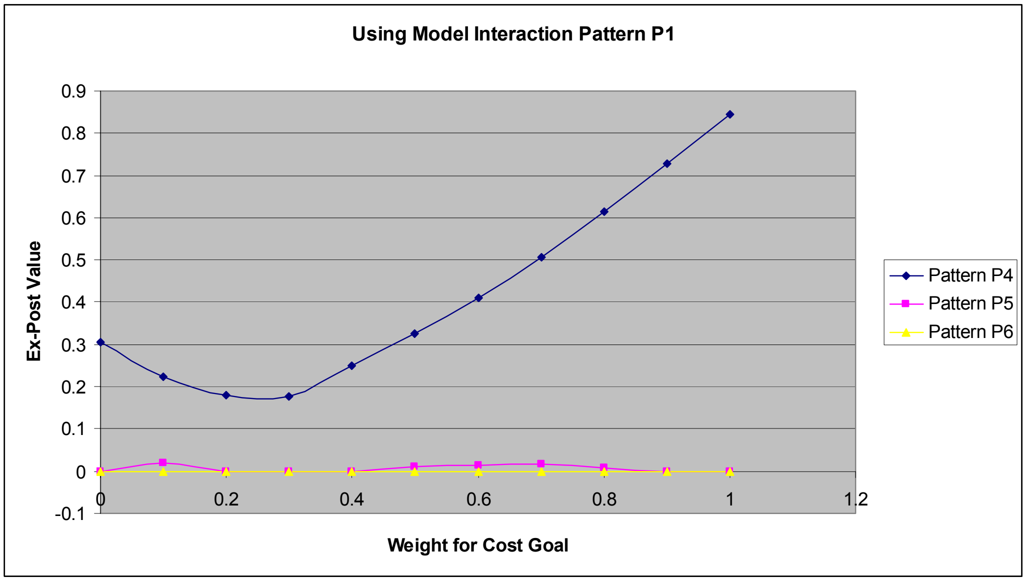


Figure 5-27 – Variation of Ex-post range for decision patterns P4, P5, and P6 with weight for cost goal – using P2 model interaction pattern

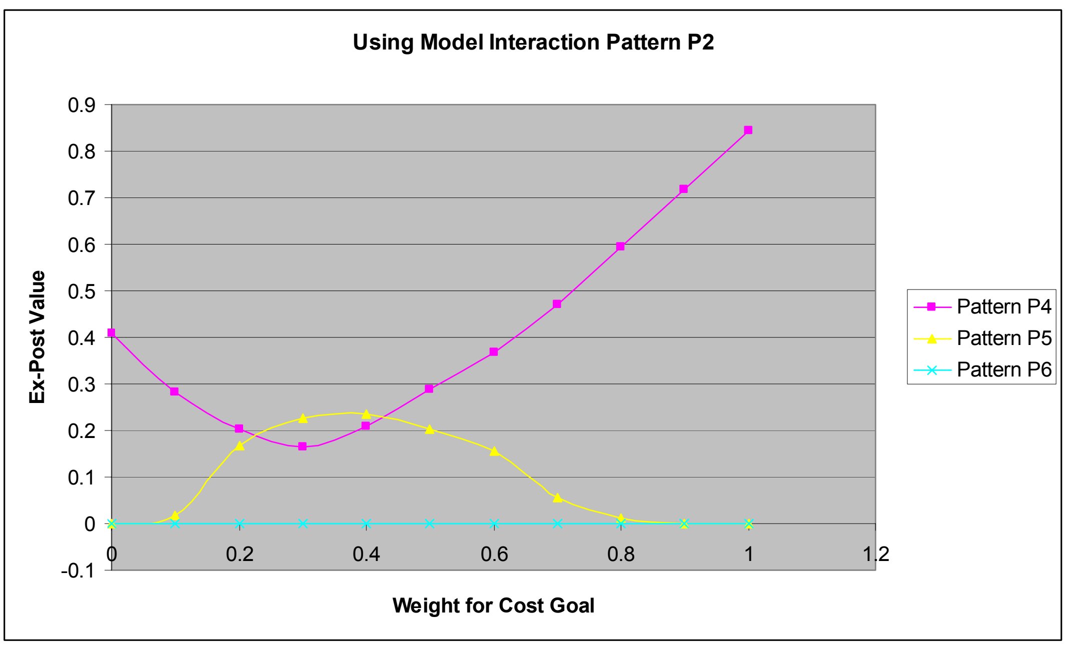


Figure 5-28 - Variation of Ex-post range for decision patterns P4, P5, and P6 with weight for cost goal – using P2 model interaction pattern

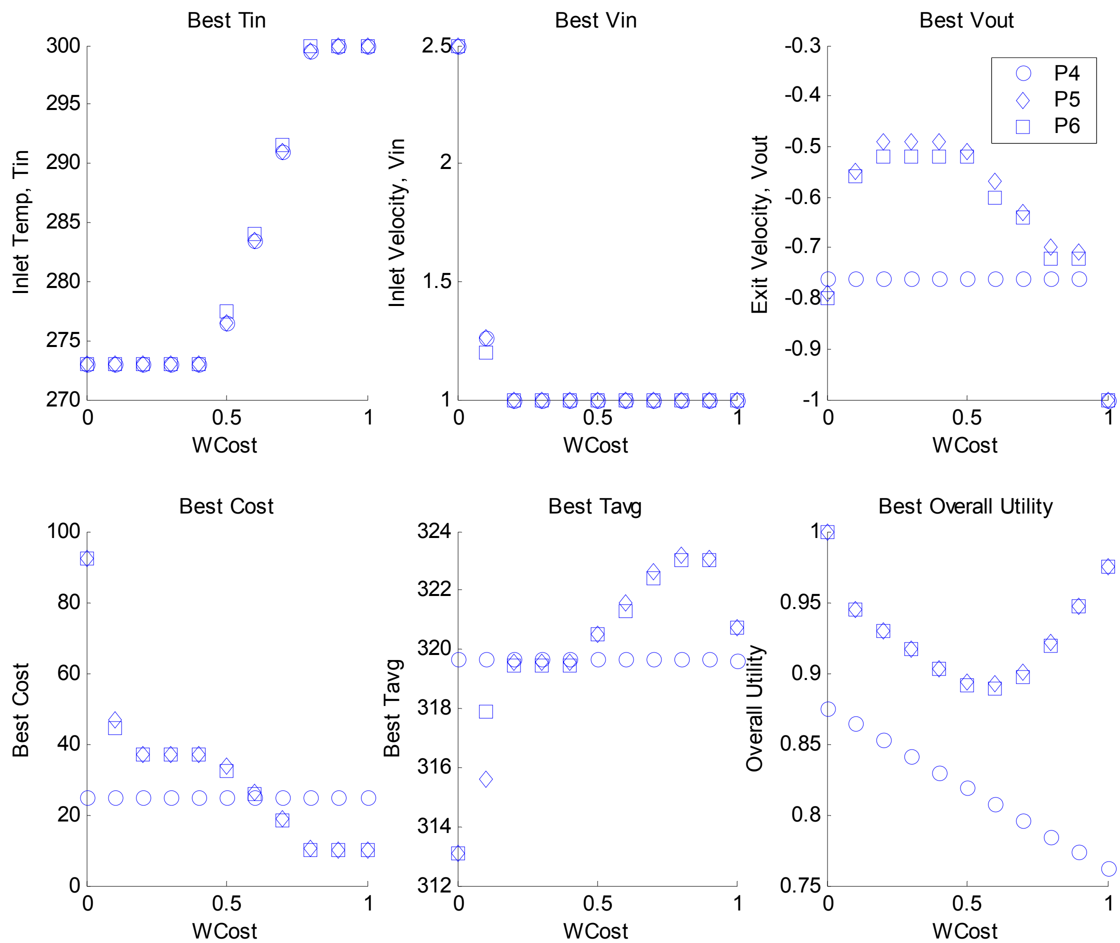


Figure 5-29 – Graphical representation of decisions made from various decision interaction patterns - using P1 model interaction pattern

Table 5-11 - Comparing the decision interaction patterns using P2 model interaction pattern

	Using Model Interaction Pattern P2									
		Decision Variables			Response Variables		Overall Utility	Value Metrics		
	Pattern	Tin	Vin	Vout	Tavg	Cost Indicator		Ex-Post Range	R1	R2
w_cost=0.0	P4	273	2.5	-0.52	320.976573	92.5	0.708613	0.408532	0.983544	0.56984
	P5	273	2.5	-0.37	312.23824	92.5	0.934394	0	1	1
	P6	273	2.5	-0.4	312.201025	92.5	0.934353	0	1	1
w_cost=0.1	P4	273	2.5	-0.52	320.976573	92.5	0.713963	0.283335	0.976144	0.538546
	P5	273	2.5	-0.34	312.32116	92.5	0.845404	0.017582	0.984318	1
	P6	273	2.5	-0.4	312.201025	92.5	0.854095	0	1	1
w_cost=0.2	P4	273	1.88	-0.52	320.976573	69.56	0.719312	0.203107	0.741025	0.52362
	P5	273	1.88	-0.31	315.821676	69.56	0.759332	0.167115	0.994262	0.855759
	P6	273	1.08	-0.12	323.249166	39.96	0.8186	0	1	1
w_cost=0.3	P4	273	1.22	-0.52	320.976573	45.14	0.724662	0.16528	0.587093	0.632298
	P5	273	1.22	-0.42	318.846062	45.14	0.760176	0.226895	0.997633	0.79136
	P6	273	1	-0.12	323.957	37	0.818799	0	1	1
w_cost=0.4	P4	273	1	-0.52	320.976573	37	0.730011	0.208067	0.621364	0.490062
	P5	273	1	-0.49	320.559025	37	0.761576	0.234103	0.988887	0.807158
	P6	273	1	-0.12	323.957	37	0.819183	0	1	1
w_cost=0.5	P4	277	1	-0.52	320.976573	3	0.73536	0.288342	0.670112	0.35227
	P5	277	1	-0.52	320.976573	33	0.780192	0.204266	0.983544	0.944747
	P6	273	1	-0.12	323.957	37	0.819567	0	1	1
w_cost=0.6	P4	287	1	-0.52	320.976573	23	0.74071	0.368976	0.861966	0.305382
	P5	287	1	-0.53	320.89827	23	0.828773	0.15542	0.988046	1
	P6	275	1	-0.12	324.695085	35	0.820302	0	1	1
w_cost=0.7	P4	296.5	1	-0.52	320.976573	13.5	0.746059	0.470313	0.999135	0.301287
	P5	296.5	1	-0.49	324.74393	13.5	0.856464	0.056106	0.931888	1
	P6	297	1	-0.56	324.480025	13	0.841745	0	1	1
w_cost=0.8	P4	300	1	-0.52	320.976573	10	0.751409	0.594599	0.99395	0.31213
	P5	300	1	-0.56	324.978345	10	0.890977	0.010359	0.959354	1
	P6	300	1	-0.64	324.008711	10	0.886141	0	1	1
w_cost=0.9	P4	300	1	-0.52	320.976573	10	0.756758	0.719017	0.997656	0.320441
	P5	300	1	-0.62	324.160781	10	0.93068	0	1	1
	P6	300	1	-0.64	324.008711	10	0.930674	0	1	1
w_cost=1.0	P4	300	1	-1	319.601	10	0.762107	0.843434	1	0.326293
	P5	300	1	-1	320.749	10	0.975207	0	1	1
	P6	300	1	-1	320.749	10	0.975207	0	1	1

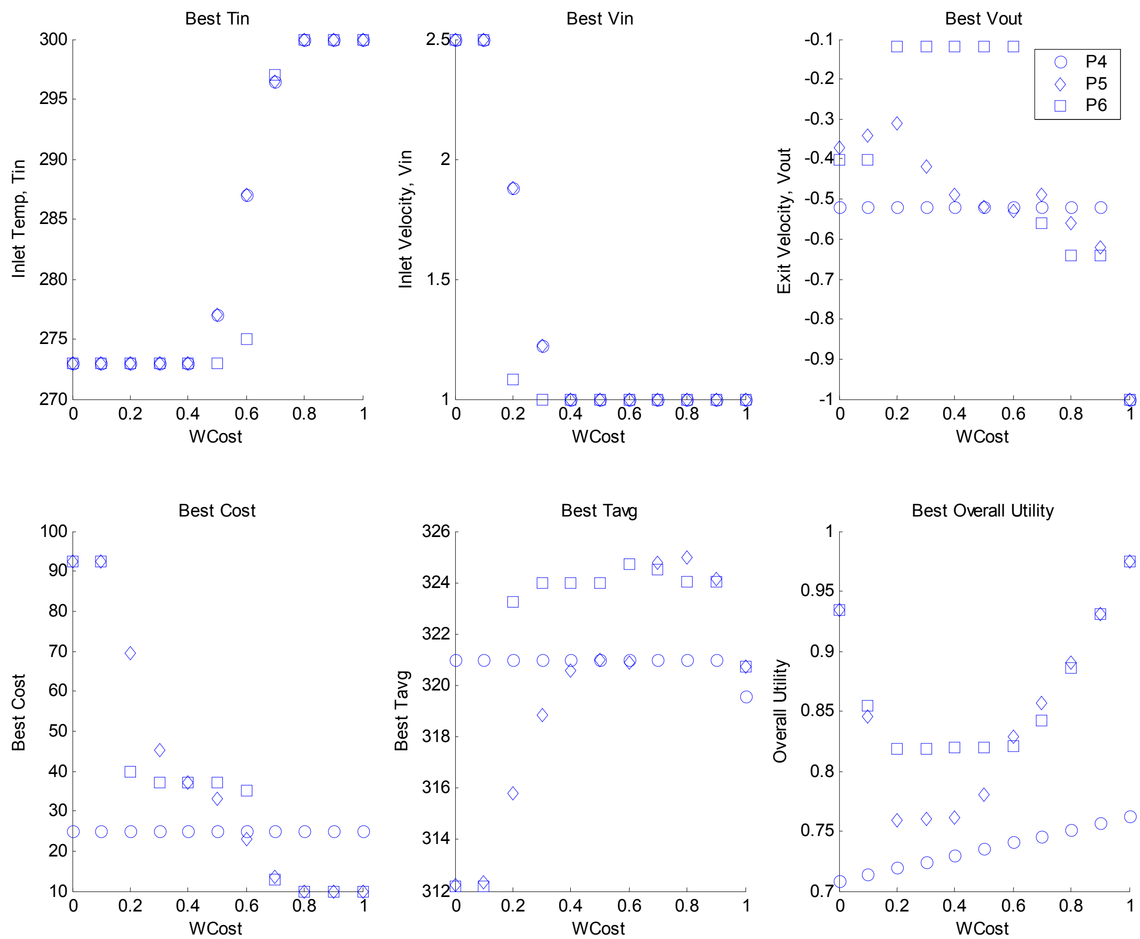


Figure 5-30 - Graphical representation of decisions made from various decision interaction patterns - using P2 model interaction pattern

Table 5-12 - Comparing the decision interaction patterns using P3 model interaction pattern

	Using Model Interaction Pattern P3									
		Decision Variables			Response Variables		Overall Utility	Value Metrics		
	Pattern	Tin	Vin	Vout	Tavg	Cost Indicator		Ex-Post Range	R1	R2
w_cost=0.0	P4	273	2.5	-0.87	319.492936	92.5	0.625028	0.168936	0.74235	0.00000
	P5	273	2.5	-0.49	312.096693	92.5	0.722582	0	1	1
	P6	273	2.5	-0.48	312.099802	92.5	0.722533	0	1	1
w_cost=0.1	P4	273	2.14	-0.87	319.492936	79.18	0.638735	0.072171	0.64459	0
	P5	273	2.14	-0.58	312.347679	79.18	0.66284	0.047235	0.938985	0
	P6	273	2.08	-0.6	313.472933	76.96	0.667957	0	1	0
w_cost=0.2	P4	273	1.46	-0.87	319.492936	54.02	0.652443	0.081538	0.832276	0
	P5	273	1.46	-0.89	319.277883	54.02	0.632563	0.087295	0.803119	0
	P6	298.5	1.78	-1	319.183598	20.47	0.66585	0	1	0
w_cost=0.3	P4	299	2.32	-0.87	319.492936	25.52	0.666151	0.173006	0.863377	0
	P5	299	2.32	-0.85	319.496178	25.52	0.710013	0.114122	0.765029	0.609542
	P6	300	1.66	-1	319.467227	16.6	0.700751	0	1	1
w_cost=0.4	P4	300	2.18	-0.87	319.492936	21.8	0.679859	0.26556	0.906774	0.120782
	P5	300	2.18	-1	319.601	21.8	0.726425	0.004704	1	1
	P6	300	1.48	-1	319.695765	14.8	0.736737	0	1	1
w_cost=0.5	P4	300	1.94	-0.87	319.492936	19.4	0.693567	0.359429	0.929247	0.193209
	P5	300	1.94	-1	319.601	19.4	0.764667	0.007368	1	1
	P6	300	1.28	-1	320.056163	12.8	0.774038	0	1	1
w_cost=0.6	P4	300	1.64	-0.87	319.492936	16.4	0.707275	0.455059	0.944815	0.238773
	P5	300	1.64	-1	319.34834	16.4	0.806033	0.010384	1	1
	P6	300	1.04	-1	320.636575	10.4	0.813101	0	1	1
w_cost=0.7	P4	300	1.28	-0.87	319.492936	12.8	0.720983	0.55213	0.963308	0.271786
	P5	300	1.28	-1	319.340291	12.8	0.851277	0.008117	1	1
	P6	300	1	-1	320.749	10	0.853605	0	1	1
w_cost=0.8	P4	300	1	-0.87	319.492936	10	0.734691	0.649231	0.979654	0.295508
	P5	300	1	-1	320.098367	10	0.895458	0.002521	1	1
	P6	300	1	-1	320.749	10	0.894139	0	1	1
w_cost=0.9	P4	300	1	-0.87	319.492936	10	0.748399	0.746333	0.991292	0.312941
	P5	300	1	-1	320.749	10	0.934673	0	1	1
	P6	300	1	-1	320.749	10	0.934673	0	1	1
w_cost=1.0	P4	300	1	-1	319.601	10	0.762107	0.843434	1	0.326293
	P5	300	1	-1	320.749	10	0.975207	0	1	1
	P6	300	1	-1	320.749	10	0.975207	0	1	1

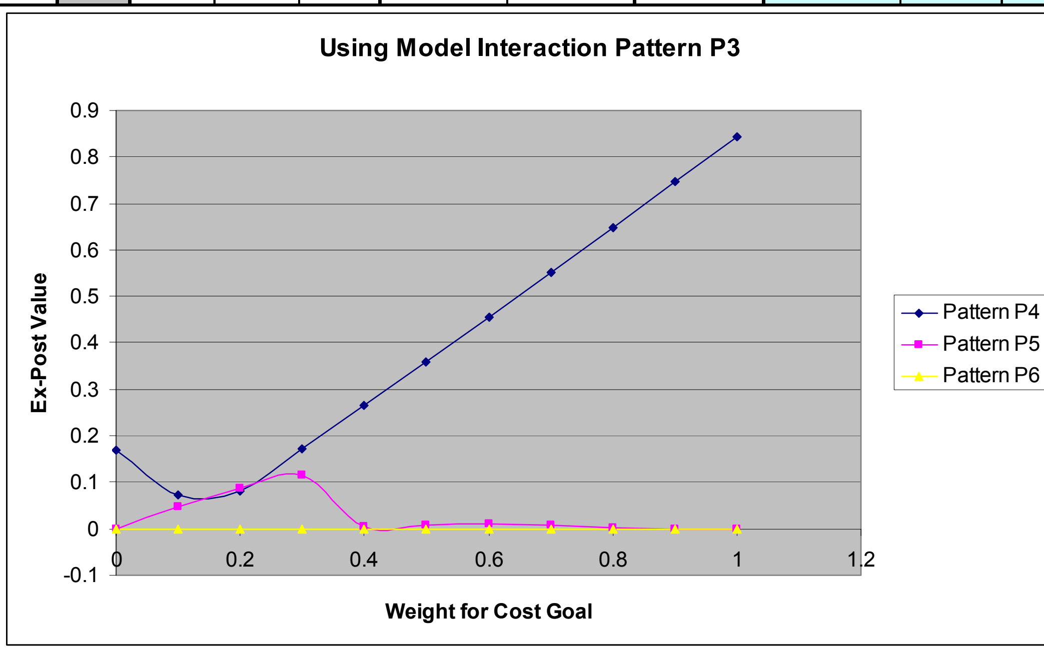


Figure 5-31 - Variation of Ex-post range for decision patterns P4, P5, and P6 with weight for cost goal – using P3 model interaction pattern

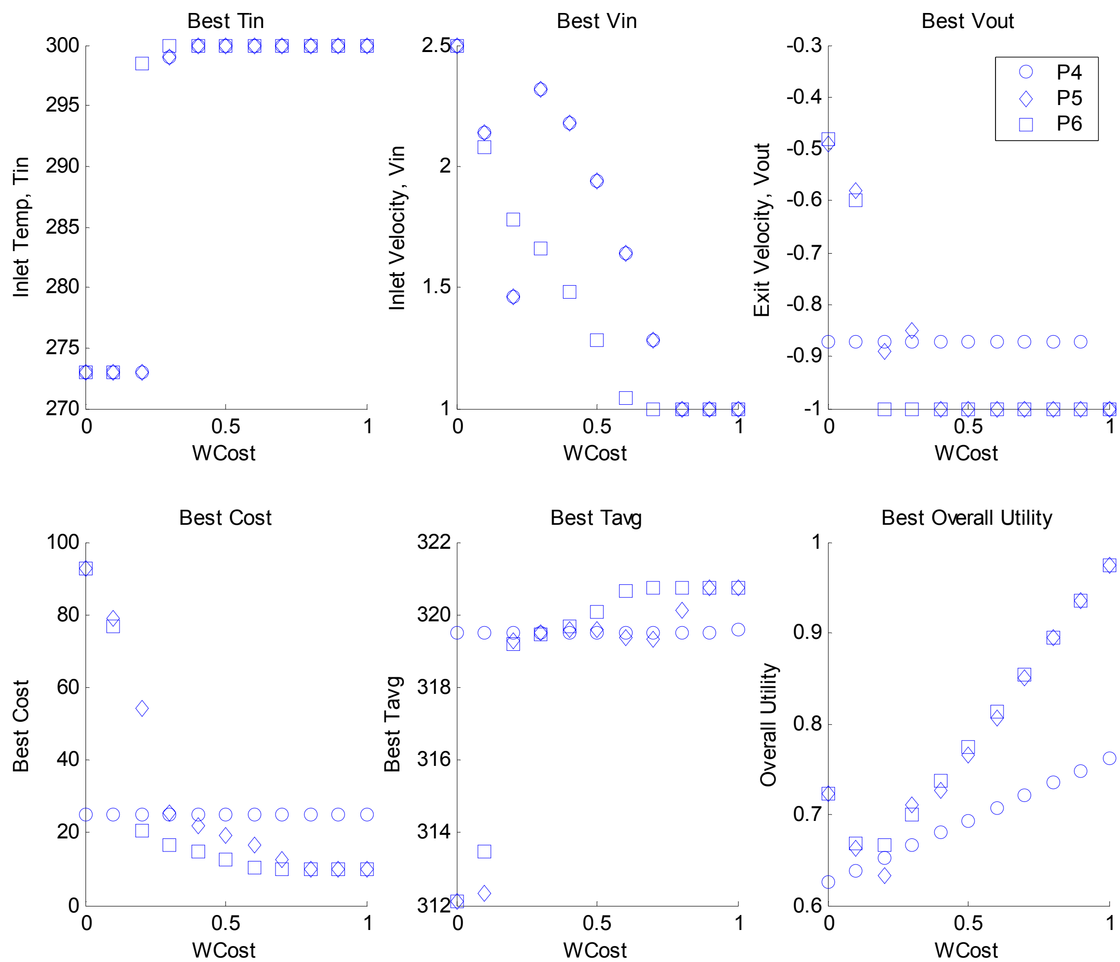


Figure 5-32 – Graphical representation of decisions made from various decision interaction patterns - using P3 model interaction pattern

As a summary, the differences in preference have a great impact on the decision interaction patterns. This is same as the observation in scale decoupling case. The point that accuracy is not the only criterion is emphasized again here. There is a difference between the information required for model decoupling and the information required for scale decoupling. In scale decoupling, the information about lower and upper bound of the outputs of models is required, whereas, in decision decoupling, the information about range of values that the design variable can take is required. It is also important to note that the difference between the decisions using three different types of model interaction patterns is due to different levels of imprecision in the simulation models. In the case of decision decoupling discussed in this section, we have not used the information about model characterization for making decisions. The decisions are made only based on the consideration of robustness to values of design variables from other decision. This is done primarily to separate the effect of coupling between decisions and the coupling between simulation models. Hence, a detailed comparison of decision decoupling results using different model interaction patterns is not carried out. In order to compare the results of decision decoupling across different model interactions, the information about error in simulation models should also be carried out. The effect of decoupling simulation models may or may not be amplified when the decisions are decoupled. Hence, the decision of model decoupling is also dependent on decision decoupling. In other words, there are two design process level (metalevel) decisions that should be made simultaneously – decision about the appropriate model interaction pattern and decision about the appropriate decision interaction pattern. The two decisions are shown in Figure 5-33. In the figure, the dependencies between the meta-level decisions and the product level decisions are shown. This indicates the need for integrated design of products and design processes. In the figure, two meta-level decisions are also shown. These two metalevel decisions are ideally coupled with each other and can be viewed as interaction pattern P6. However, depending on the strength of this coupling, the decisions can be simplified into sequential or independent decision patterns – P4 and P5 respectively. Hence, the same idea of decision decoupling can also be applied to design process related decisions. This issue of simplification of meta-level decisions is not discussed in this dissertation and is left at this point as an opportunity for future work.

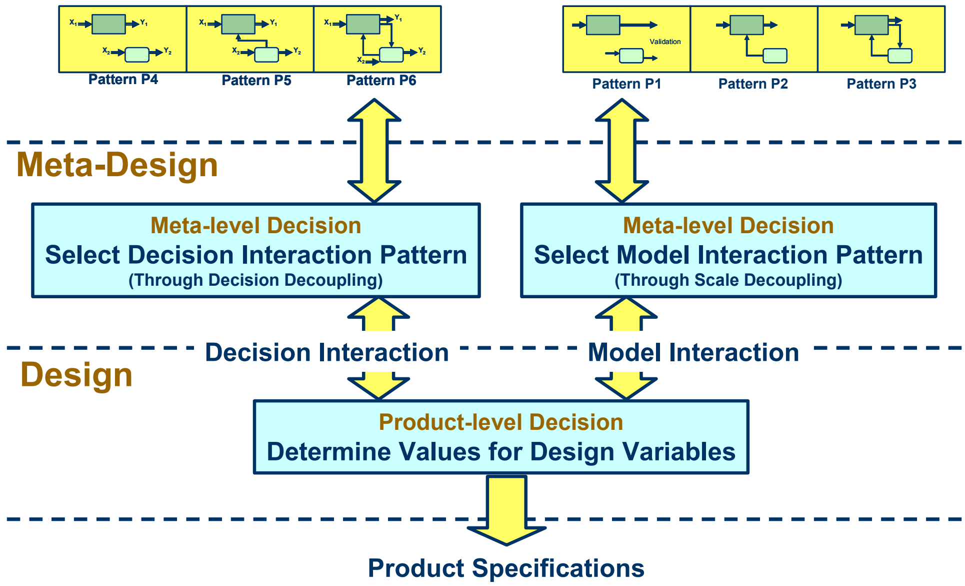


Figure 5-33 – Integrated design and meta-design as supported by the proposed design framework

Implications on Modeling Design Information

The simplification of interaction patterns discussed in this chapter show that different decision patterns are suitable for different decision making scenarios. Even in the case of single product design, changing the designers' preferences results in different interaction patterns. As discussed in Chapter 3, different design processes can be used for different kinds of interaction patterns. For example, an independent decision pattern can be executed by different kinds of algorithms such as exhaustive search, gradient based search, approximation based methods, etc. The coupled decisions can be solved by combination of decisions into a single decision, sequential iterative method, set based focalization method (discussed in Chapter 6), or by intersection of rational reaction sets. This means that the suitable design processes for a design scenario changes drastically by merely changing some preferences in the design decision. This imposes a strong requirement for the computational framework to be used for simulation-based design.

The computational framework should support designers to rapidly change and use design processes for a design problem, which requires separation of declarative decision specific information from the process used to execute the decision. The second requirement that arises from the discussion in this section is that of composability of design processes from different decisions. This is because the design processes associated with two decisions may either be used separately (if they are used in interaction pattern P4) or used in a coupled fashion (if they are used in interaction pattern P6). During the design process exploration, the designers may shift from one type of process to another. Hence, the design processes associated with individual decisions should be composable. These two requirements are discussed in detail in Chapter 7 . A design information modeling strategy to address these requirements is presented in Chapter 8.

5.5 On Verification and Validation

The methods for simplification of design processes are geared towards addressing the second research question (RQ2) in this dissertation – "*How should multiscale design processes be systematically simplified and models refined in a targeted manner to support quick design decision making without compromising the decision quality?*" The aspect of model refinement is addressed in Chapter 4. The answer to this research question is supported by two hypotheses: (H2.1) design processes can be simplified and models refined by making tradeoffs between value of information obtained via simulations and need to achieve robust, satisficing solutions, and (H2.2) design processes can be simplified using decoupling of scales, decisions and functionalities. In this chapter, hypothesis H2.1 is embodied in the methods for simplification (see Sections 5.3.1 and 5.4.1), where we show how the value of information can be used to make design process decision relating to appropriate level of simplification of design processes. A part of hypothesis H2.2 is embodied in this chapter by developing methods for decoupling of scales and decisions. Functional decoupling is discussed in the next chapter. The validation section in this chapter is focused on the methods for simplification of design processes through scale and decision decoupling. Since the methods are based on the value of information metric that is discussed in detail in Chapter 4, theoretical structural validation is already performed. Two aspects of the validation square are addressed Empirical Structural Validation and Empirical Performance Validation in this Section. An overview of the validation performed of scale and decision decoupling methods discussed in this chapter is provided in Figure 5-34. The figure provides Chapter 5 specific details to the overall validation strategy for the dissertation, which is presented in Figure 1-12. This validation square corresponds to one of the validation subsquares presented in Figure 1-10.

Empirical structural validation involves accepting the appropriateness of the example problems used to verify the performance of the method. The example problem used in this chapter is a multiscale datacenter cooling system design problem. The example is appropriate for validating the methods for scale decoupling because it can be formulated as a single decision with multiple simulation models feeding information for decision making (patterns P1, P2, and P3). Further, the lower and upper bounds of possible values taken by the response values can also be calculated from the knowledge about 'actual' behavior of the system. Ideally, the actual behavior of the system should be determined from the experiments. However, for the purpose of validating the method, we treat the completely coupled model as the exact model. From this information, the bounds on values from independent and sequential patterns are evaluated. This is important for validating scale decoupling method. The problem is suitable for validating the method for decision decoupling because it can be formulated as a multiple decisions that are coupled with each other. Hence, it can be modeled as interaction patterns P4, P5 and P6. The information required for decision decoupling includes bounds on design variables, which is available. The problem is also suitable because it is demonstrates the aspects of integrated design of products and design processes.

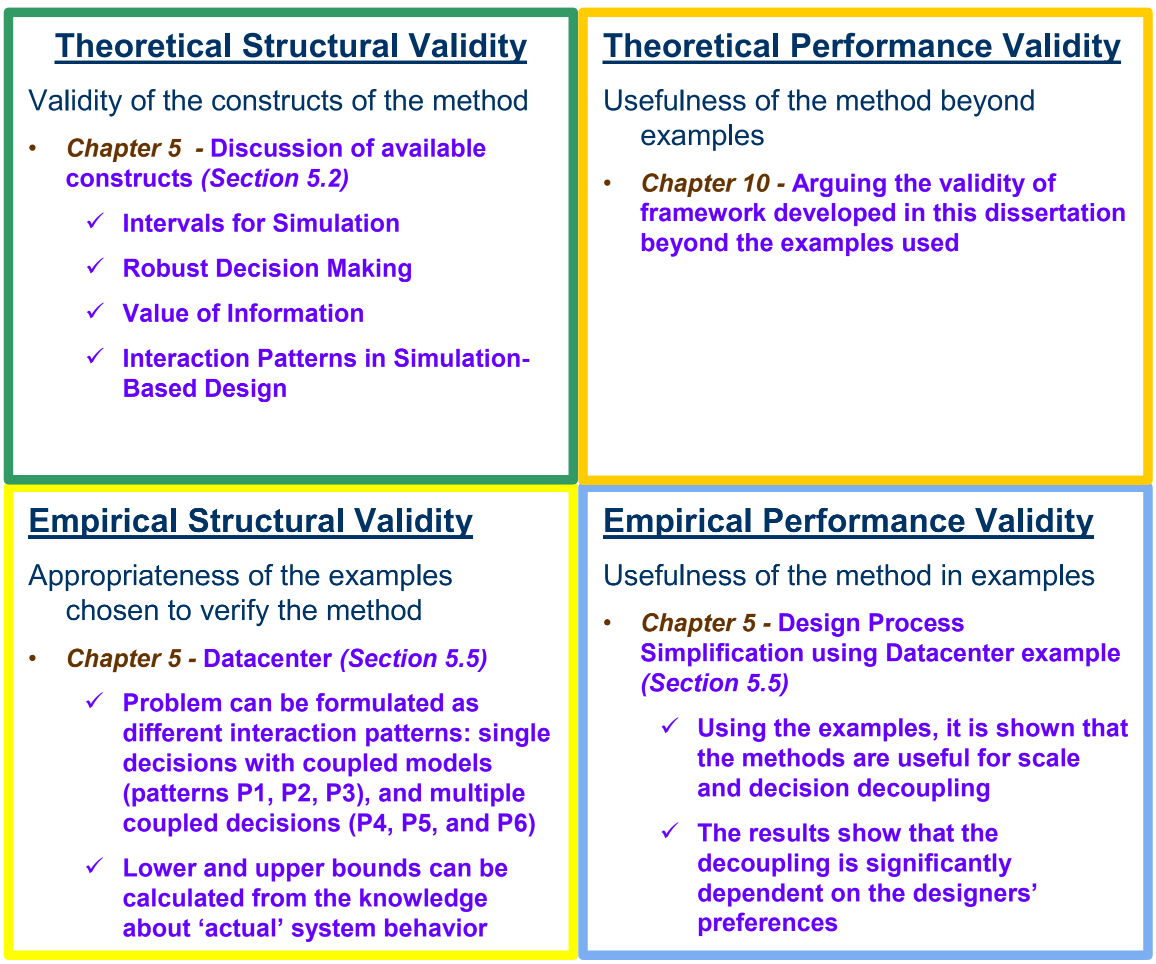


Figure 5-34 – Validation of methods for scale and decision decoupling addressed in Chapter 5

Empirical performance validation consists of accepting the usefulness of the outcome with respect to the initial purpose and accepting that the achieved usefulness is related to applying the method. In this chapter, it is shown that the methods can be used for scale and decision decoupling. The results from scale decoupling indicate that although the models are inherently coupled, they can be modeled with sequential information flow, which results in savings of computational time. The results from decision decoupling indicate that although the decisions are coupled with each other, in some preference scenarios, they can be modeled as sequential decisions. In other scenarios, they should be modeled in a coupled fashion. This process level decision is made by using methods presented in this chapter.

5.6 Role of Chapter 5 in this Dissertation

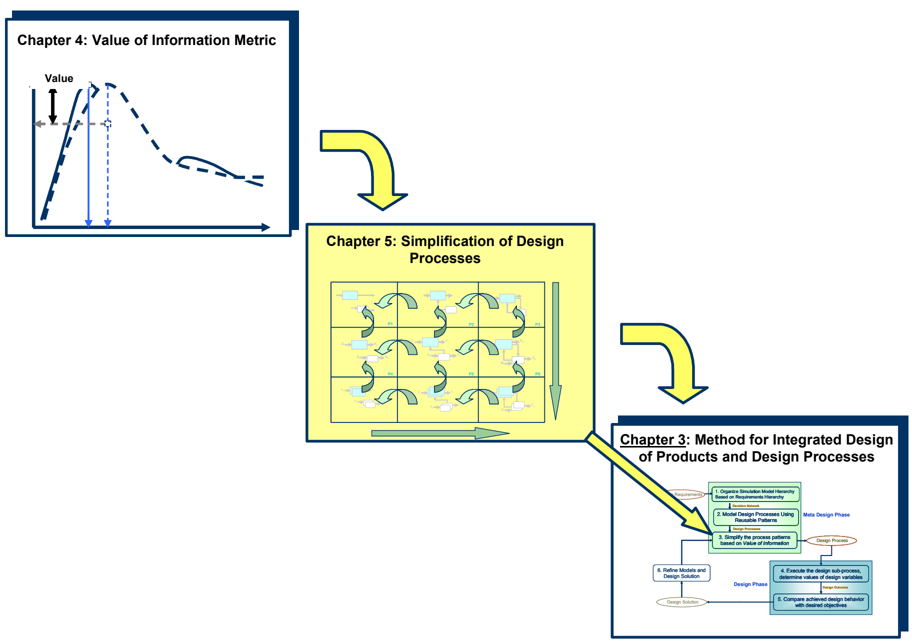


Figure 5-35 - Relationship of Chapter 5 with other chapters in the dissertation

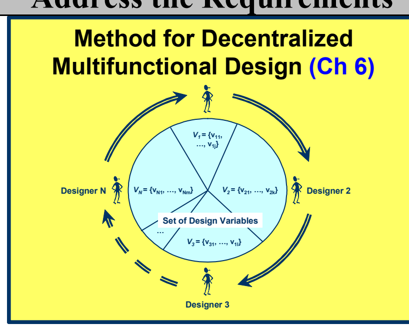
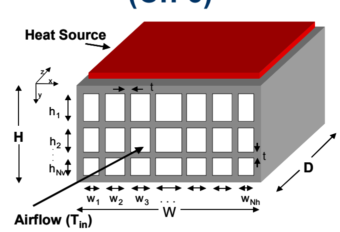
In this chapter, details of one of the steps in the design method discussed in Chapter 3 are discussed. This step is related to systematic simplification of design processes. The value of information metric developed in the previous chapter (Chapter 4) is used to make design process simplification related decisions. The relationship of this chapter with Chapter 3 and Chapter 4 is shown in Figure 5-35.

Specifically, in this chapter, the simplification of design processes is carried out for two types of interaction patterns – model interactions and decision interactions. The simplification associated with these two types of patterns is referred to scale decoupling and decision decoupling respectively. The following chapter (Chapter 6) deals with the third type of interaction patterns that is defined in terms of functional coupling. Functional coupling is an important aspect of multifunctional design where different distributed designers are in charge of satisfying different functional requirements. In contrast to this chapter, where the focus is on methods that support decoupling of design processes, the primary focus in the next chapter (Chapter 6) is on decision making in the presence of functional coupling. Only a short subsection is devoted to decoupling functionally coupled decisions.

Chapter 6 Functional Decoupling in Collaborative Multifunctional Design using Set-Based Methods

In this chapter, we address the second requirement for the framework for integrated design of products and design processes – "Support for decentralized, multifunctional design". The highlights of this chapter are presented in Table 6-1, which is a subset of the framework components presented in Table 1-6. The aspects of the design framework discussed in the Chapters 1 through 5 do not involve explicit consideration of distributed nature of multiple designers. The distribution of experts and computational requirements requires that the communication of information between designers be consise and systematic. Excessive iterations between designers should be avoided. These challenges associated with the collaborative multifunctional design scenarios is addressed in this chapter through an interval-based (or set-based) method. The method is validated using a multifunctional Linear Cellular Alloy (LCA) design problem.

Table 6-1 – Highlights of the framework requirements, components of the framework, and the validation example presented in Chapter 6

Framework Requirements	Components of the Framework Developed to Address the Requirements	Validation Examples
2) Support for decentralized, multifunctional design		**LCA Design Example (Ch 6)**  ***Purpose:*** To validate the interval-based focalization method

Specifically, the focus is on multifunctional design, where different designers are responsible for achieving different functional requirements. Hence, the emphasis is on interaction patterns P7 through P9 discussed in Section 3.5.2. This is in contrast to the previous chapters where a single designer wants to achieve monofunctional requirements (patterns P4 through P6). In Chapter 5, the discussion is primarily centered on decoupling decisions and scales. However, in this chapter, we only provide a brief discussion of functional decoupling, which can be carried out if the coupling between different functional behaviors is weak. This is because the method for functional decoupling is similar to that for scale and decision decoupling. A more common and interesting scenario is where functional decoupling cannot be performed. Most of this chapter is devoted to such scenarios with strong functional coupling. Another distinction between the methods presented in this chapter and those presented in Chapter 5 is that the methods presented here are developed for multiple designers collaborating together to design the product for different functional aspects.

The weak coupling case is presented in Section 6.1. An overview of the strongly coupled scenario and a review of relevant literature are provided in Section 6.2. The limitations of existing approaches for decentralized multifunctional design methods are presented. These limitations provide motivation for developing a method for design in such scenarios. In Section 6.3, we present two theoretical constructs used in this chapter – game theory protocols for collaborative design, and box consistency. In Section 6.4, a new interval-based focalization method is presented for decentralized multifunctional design. This method addresses the limitations of existing methods identified in Section 6.2. The method is illustrated using two example problems – simple quadratic responses (in Sections 6.4.1) and Linear Cellular Alloy design (in Section 6.4.3). The effect of initial conditions and convergence criterion is provided in Sections 6.4.2 and 6.4.4

respectively. Finally, a discussion of verification and validation of the method developed in this chapter is presented in 6.5. The aspects of research addressed in this chapter are highlighted in Figure 6-1. The validation example presented in this chapter is that of a design of a multifunctional Linear Cellular Alloy.

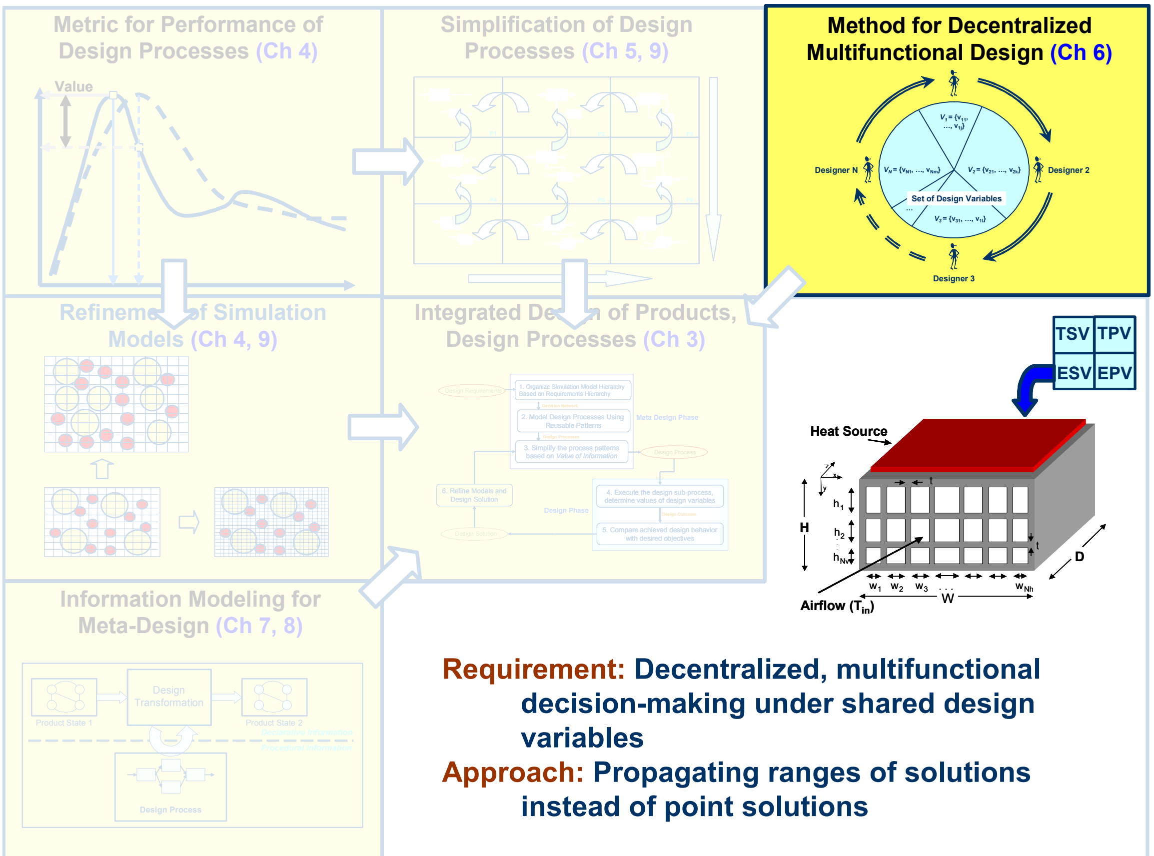


Figure 6-1 – Research aspects highlighted in Chapter 6

6.1 Eliminating Weak Functional Couplings

In the case of functional coupling, the design problem is characterized by a common set of design variables shared between different designers. The same set of design variables has an impact on multiple functional characteristics. For example, consider a materials design problem for energetic-structural materials where the material should have sufficient strength to bear loads and should also be able to release energy under predefined conditions. In this problem both the functional characteristics – strength and energy release are dependent on a common set of design variables such as the constituent volume fractions, size of constituent particles, and their distribution in space. Since the same sets of design variables control different functional behavior, the system (material, in this case) is functionally coupled, and cannot be designed separately for individual functional requirements. Generally, in multiscale multifunctional problems, the coupling is between decisions in not only because of shared design variables, but also due to the interaction between physical phenomena (e.g., structural and energy release).

The coupling between decisions is categorized into two types based on its strength – weak or strong coupling. As the name implies, weak coupling does not have a major impact on the designers' decision, whereas strong couplings have a large impact. Note that analogous to the value of information, the couplings are defined in terms of impact on decisions rather than the error in behavior prediction. Hence, if the decisions are weakly coupled, they can be considered individually. Again the metric that can be used to determine whether decisions are strongly or weakly coupled is value of information discussed in Chapter 4. Note that the decisions made after decoupling functionalities should be based on the ideas of robust design. The approach for decoupling decisions related to different functionalities (patterns P7 through P9) is same as the approach for decoupling decisions at different scales. Due to this similarity, we do not consider weak coupling in detail in this chapter. The focus is mainly on strong coupling between decisions and is discussed in the remaining part of this chapter.

6.2 Separating Strongly Coupled Factors (Simplification through Functional Decoupling) – Interval Based Focalization Approach

Multi-functional design problems are characterized by strong coupling between design variables that are controlled by stakeholders from different disciplines. This coupling necessitates efficient modeling of interactions between multiple designers who want to achieve conflicting objectives but share control over design variables. Various game-theoretic protocols such as cooperative, non-cooperative, and leader/follower have been used to model interactions between designers. Non-cooperative game theory protocols are of particular interest for modeling cooperation in multi-functional design problems. These are focused upon in this chapter because they more closely reflect the level of information exchange possible in a distributed environment. Two strategies for solving such non-cooperative game theory problems are – a) passing Rational Reaction Sets (RRS) among designers and combining these to find points of intersection and b) exchanging single points in the design space in an iterative fashion until the solution converges to a single point. While the first strategy is computationally expensive because it requires each designer to consider all possible outcomes of decisions made by other designers, the second strategy may result in divergence of the solution.

In order to overcome these problems, we present an interval-based focalization method for executing decentralized decision-making problems that are common to multi-functional design scenarios. ***The method involves propagating ranges of design variables and systematically eliminating infeasible portions of the shared design space.*** This stands in marked contrast to the successive consideration of single points, as emphasized in current multifunctional design methods. The key advantages of the proposed method are *a)* targeted reduction of design freedom and *b)* non divergence of solutions. The method is illustrated using two sample scenarios – solution of a decision problem with quadratic objectives and design of multi-functional Linear Cellular Alloys (LCAs). Implications include use of the method to guide design space partitioning and control assignment. Notice that set-based design is an old concept. It has been conceptually employed in the context of engineering design by various researchers such as Ward, Liker, and Sobek (Ward, Liker et al. 1995; Liker, Sobek et al. 1996; Sobek and Ward 1996). In this chapter, we present one specific embodiment of the set-based design for multifunctional applications.

Imagine a complex design scenario such as the design of a multi-functional, multi-scale product/material system with numerous, conflicting requirements. One of the characteristics of such a multi-functional design problem is that experts from different domains must work together both in a serial and parallel fashion in order to achieve their individual as well as overarching system level goals. For example, the product may be required to simultaneously meet structural and thermal requirements, while satisfying geometric constraints. It is in this regard that experts from relevant domains (i.e., structural, thermal, manufacturing, etc.) are called upon; they are required to contribute their respective expertise and collaborate in order to accomplish their individual and common goals. In multi-functional design scenarios such as this, effectiveness of collaboration between designers is the key to success. The problem of choosing a method for effective collaboration is essentially that of finding the most appropriate way of utilizing knowledge initially dispersed among domain experts, subject to organizational barriers and process dynamics.

Depending on the nature of the underlying design process, there are two collaboration strategies that are commonly employed for effectively synthesizing contributions of interacting designers. The first strategy is based on *centralized decision-making* and requires a single transfer of knowledge from various domain experts to a central decision-maker. It is based on gathering and consolidating information and facilitates the attainment of Pareto-optimal solutions via simultaneous consideration of system level tradeoffs. However, even slight changes in any of the design goals or requirements pertaining to the integrated domains or the design environment may render the gathered knowledge incomplete; optimal solutions may no longer be obtainable. Specifically, the centralized decision-maker may not have the required expertise to adjust domain models in order to accommodate the required changes, rendering iteration with other stakeholders unavoidable. Hayek (Hayek 1945), on the other hand, advocates *decentralized decision-making,* pointing out that it is important to delegate responsibility to persons "on the spot" who have intimate knowledge of their respective domains and are (consequently) capable of making any required inferences. Lee and Whang (Lee and Whang 1999) present decentralized decision-making methods in the context of supply chains, whereas Chanron and co-authors (Chanron, Singh et al. 2004) offer a decentralized decision-making strategy for the solution of engineering design problems.

Decentralized decisions can be classified as being either *coupled* or *decoupled* in nature. *Decoupled decisions* are characterized by independence in formulation and solution, thereby allowing a unidirectional (sequential) flow of information and greatly facilitating the underlying design processes. *Coupled decisions*, on the other hand, are more complex and require a two-way flow of information between decision-makers as well as active involvement of domain experts throughout the decision-making process. *Coupled* decisions are especially significant in multi-functional design problems, where different designers and domain experts control a common set of design variables and share responsibility for achieving different objectives. Considering the prevalence of coupled decisions in engineering design in general and within multi-functional design in particular, we present an interval-based technique for their resolution in this chapter. Before proceeding, however, we underscore some of the nuances inherent in the solution of *coupled* problems as well as current means of resolution at the hand of a simple problem, requiring the interaction of two designers.

Consider the scenario shown in Figure 6-2, where Designers A and B are responsible for achieving their respective design objectives. These objectives are defined in terms of the maximization, minimization, or matching of *response variables* - **Y**. In the given scenario Designer A controls a set of *design variables* X_A while Designer B controls a set of design variables X_B. Since the two decisions are coupled, Designer A cannot make a decision about X_A unless the values of X_B are fixed. Similarly, Designer B cannot make a determination with regard to X_B unless the values of X_A are known. One of the strategies commonly implemented for solving such a coupled, decentralized problem is point-based iteration. In point-based iteration interacting designers consider a single point within a given design space at a time and iteratively adjust this point until they converge on a solution that satisfies their respective design objectives (which are functions of response variables). Procedurally, one of the designers (say Designer A in the scenario depicted in Figure 6-2) starts by assuming values of design variables controlled by the other designer (X_B) and determines values for his/her design variables

(X_A) so that his/her objectives (Y_A) are satisfied. Using these values of design variables (X_A), Designer B can then determine suitable values for his/her design variables (X_B) considering his/her own objectives (Y_B). This process continues until converging to a single point in the design space (X_A, X_B).

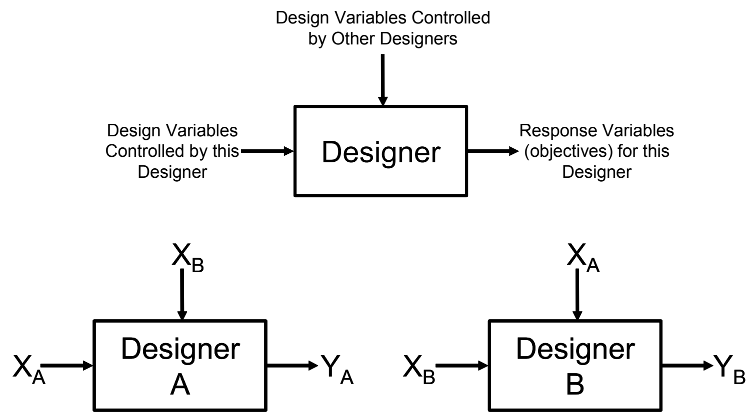


Figure 6-2 – A two-designer scenario for decomposing strongly coupled system

Design Freedom is defined as the ***extent to which a system can be adjusted while still meeting the design requirements posed for it*** (Simpson, Rosen et al. 1998). Disadvantages of a point-based iterative method relate to the manner in which *a)* design freedom is reduced and *b)* convergence is achieved. The first disadvantage is that design freedom is reduced from the initial ranges of design variables to point values in a single step. This severely limits designers in accommodating any changes in requirements. A more gradual and systematic reduction of the design space and the associated design freedom, on the other hand, reduces premature (unnecessary) elimination of potential solutions. This is illustrated in Figure 6-3. In this figure, a comparison of point-based (see Figure 6-3a) and interval-based methods (see Figure 6-3b) is presented in terms of *a)* design space, made up of the design variables under the designers' control (X_A, X_B), *b)* response space, which constitutes the response variables (Y_A, Y_B) and *c)* the associated design freedom. The numbers on the figure represent successive exchanges among interacting designers and the associated effects on the design space, the response space and the associated design freedom. The arrows in the point-based approach denote the progression of the design process by moving from point to point in the design space. The rectangles in the design and response spaces of the interval-based approach, on the other hand, refer to regions under consideration at given points in time.

In order to more effectively manage design freedom throughout the design process, a number of set-based design techniques have been proposed for application in design (Ward, Liker et al. 1995; Liker, Sobek et al. 1996; Sobek and Ward 1996). The primary purpose for using such set-based design methods is (1) the communication of sets of possibilities and (2) the subsequent narrowing of these sets, balancing the need to gain more knowledge and progressively reduce uncertainty (Sobek and Ward 1996). These sets of possibilities are represented by a series of rectangles, the areas of which decrease with successive iterations, in Figure 6-3. As shown, the reduction in design freedom for point based methods occurs in a single step, whereas that associated with interval-based methods is more gradual. An additional advantage, particular to set-based methods is the ability to make simultaneous progress on interdependent design problems and increase their concurrency, without reformulation as a single design problem. In this chapter, *we build on the concept of set-based design through the implementation of interval arithmetic.* Instead of communicating information about a single point in the design space at a time, we advocate the transfer of feasible ranges of values for given design variables. The key advantage of such an interaction mechanism is that design freedom remains open for a longer period of time, thereby accommodating changes in the requirements during the execution of the design process and maintaining the autonomy of experts over their respective domains. Interval arithmetic has been used for modeling selection decisions in (Reddy and Mistree 1992).

The second limitation of point-based methods is that the resulting solutions may be unstable. Additionally, results may never converge to the problem's Nash equilibrium. Chanron and co-authors (Chanron and Lewis 2003; Chanron and Lewis 2004; Chanron, Singh et al. 2004) investigate the underlying dynamics of decentralized processes and corresponding convergence and stability criteria using numerical series and linear algebra. Their investigation, however, is based on the assumption that the system has previously been decomposed. Chanron and co-authors do not investigate the effect of decomposition strategies on convergence. In this chapter, *we illustrate the effect of different decomposition strategies on problem convergence characteristics* and *offer mathematical criteria for system decomposition*.

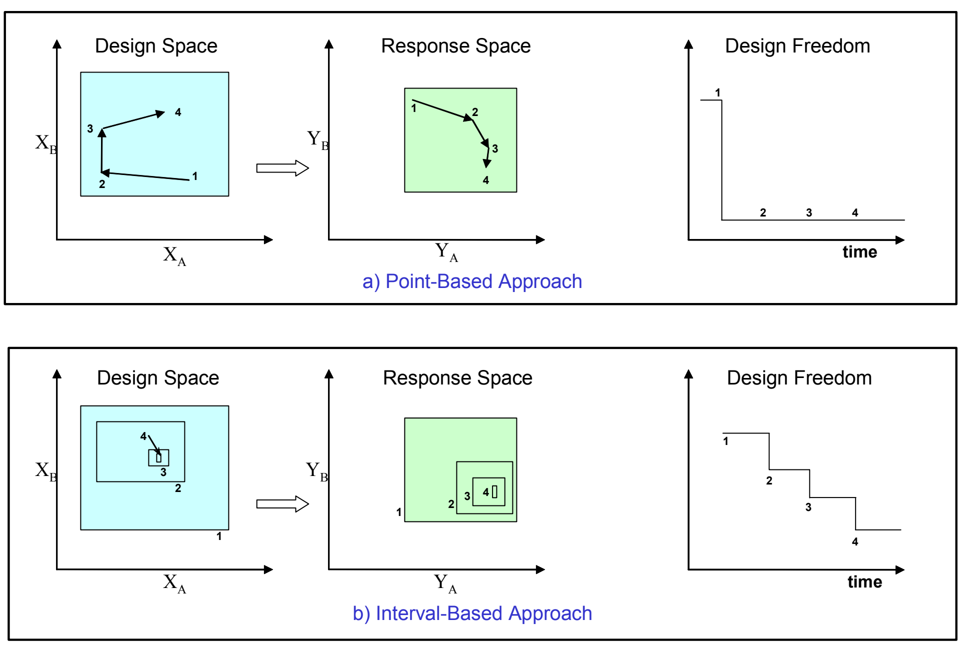


Figure 6-3 - Comparison of point-based and interval based methods for decision making

In summary, we present a design method that aids system level designers in executing design processes multi-functional product/material systems, where designers in charge of different functional requirements share a common set of design variables. Using this method, design freedom is reduced only when eliminating infeasible range of a design space, thereby accommodating unforeseen changes in design objectives over time. The key advantages of this method are that *a)* the resulting, decomposed system never diverges and *b)* design freedom is reduced systematically (though not prematurely) throughout the design process. With this in mind, we provide an overview of non-cooperative game theoretic protocols for modeling interactions between designers in Section 6.3. We also present the principle of Box Consistency – a mathematical tool emanating from Interval Arithmetic – which serves as a foundation for the proposed interval-based focalization method. We proceed to outline our method for multi-functional design in Section 6.4 and illustrate this method at the hand of a non-linear, multifunctional design example in Section 6.4.3. The convergence characteristics of this method are described in Section 6.4.4. Finally, we provide a discussion with regard to current limitations of the proposed method and future opportunities for extension in Section 6.5.

6.3 Theoretical Constructs Used in this Chapter

There are a number of different mechanisms commonly employed for decentralized decision-making in multi-functional design problems. These include applications of multi-disciplinary optimization approaches (see e.g., (Balling and Sobieski 1996)), negotiations (see, e.g., (Kusiak, Wang et al. 1996; Scott and Antonsson 1996; Scott 1999)), and finally game theoretic principles (see e.g., (Rao 1987; Lewis and Mistree

1997; Marston and Mistree 2000; Xiao, Zeng et al. 2002)). Since game theory has been formalized for both centralized and de-centralized decision-making, we build on the underlying protocols to develop the interval-based focalization method proposed in this chapter. An overview of game-theory as applied within the field of engineering design is provided in Section 6.3.1, with an emphasis on the *non-cooperative formulation* that appropriately represents coupled, decentralized decision-making. In order to develop the *solution mechanism* for this problem formulation, we rely on *Box Consistency,* a mathematical construct developed within the area of interval arithmetic. A detailed discussion of Box Consistency follows in Section 6.3.2.

6.3.1 Game Theory Protocols for Collaborative Design

Game theory has been employed as a means of conflict resolution in engineering design, with instantiations varying depending on the nature of the underlying problem addressed. Myerson (Myerson 1991), for example, presents game theory as a method for resolving conflict between multiple decision-makers controlling subsets of design variables and striving to minimize individual cost functions. Rao and Freiheit (Rao and Freihet 1991) present a modified game theory method to solve multi-objective problems, that is subsequently extended by Rao (Rao 1987) for structural optimization and by Badhrinath and Rao (Badhrinath and Rao 1996) for the integrated design of control structure. Hacker and Lewis (Hacker and Lewis 1998) develop a robust design method to reduce elements of uncertainty in a non-cooperative system that result from prediction of disciplinary subsystem behavior. This uncertainty is due predominantly to a lack of global control. Unknown and uncontrollable design decisions (made within competing subsystems) are thus modeled as internal noise variables via the application of Robust

Design in conjunction with game theoretic protocols. The goal is to reduce the effect of interacting decision-makers on one another. Subsequently, Kalsi, Hacker, and Lewis (Kalsi, Hacker et al. 1999) proceed to build upon this framework by solving disciplinary sub-problems independently from the rest of the system through the incorporation of ranges. Changes in control variables are also considered explicitly, thereby including Type II Robust Design principles. Hernández implements game theoretic principles to establish a mathematically supported cooperative framework that enhances the practical, effective, and efficient integration of the enterprise design process (Hernández 1998). Specifically, Hernández provides a method, appropriate for the formulation and solution of design problems in a manner consistent with this framework, where enterprise decisions are coordinated through a design formulation based on the game theoretical formulation of the enterprise design process. In later work Hernández (Hernández, Seepersad et al. 2002; Hernández, Seepersad et al. 2002) formalizes the interactions of two collaborating stakeholders in engineering design processes. Marston (Marston, Allen et al. 2000; Marston and Mistree 2000) develops a multi-designer model of engineering design that accounts for uncertainty, cooperation, non-cooperation, and coalitions, using the mathematics of decision and game theory. In doing so he introduces the notion of Game-Based Design as "…the set of mathematically complete principles of rational behavior for designers in any design scenario" in Ref. (Marston 2000).

Lewis and Mistree (Lewis and Mistree 1997) abstract the mathematical foundations of game theory to model complex design processes. They model the strategic relationships among designers sharing a common design space using game theoretic principles and identify *Pareto Cooperation, Stackelberg Leader/Follower,* and *Nash*

Non-Cooperation as the three game theoretic protocols most representative of the interactions required for decentralized design. *Pareto Cooperation* is employed to represent *centralized* decision making, where all required information is available to every collaborating designer. A *Pareto optimal* solution is achieved when no single designer can improve his or her performance without negatively affecting that of another. *Stackelberg Leader/Follower* protocols are implemented to model sequential decision making processes where the "leader" makes his or her decision, based on the assumption that the "follower" will behave rationally. The follower then makes his or her decision within the constraints emanating form the leader's choice.

Nash Non-Cooperation refers to decentralized decision processes where designers have to make decisions in isolation due to organizational barriers, time schedules, and geographical constraints. It is focused on formulation of strategies that "rational" individuals follow when their actions and objectives are affected by others, its mathematical models are suitable for formulating decisions in collaborative design (Hernández, Seepersad et al. 2002). The *Nash Non-Cooperative* protocol is particularly important in multi-functional design scenarios because of the non-required collocation of design experts and extensive coupling within the design space.

In *Nash Non-Cooperative* protocols, decision-makers formulate Rational Reaction Sets (RRS) or Best Reply Correspondences (BRC). A RRS is a mapping (either a mathematical or a fitted function) that relates the values of design variables under a designer's control to values of design variables controlled by other stakeholders. For example, in a two designer scenario where the first designer controls design variable set X_A and the second designer controls variable set X_B, the RRSs of the first designer is given by $(X_A)_{RRS} = f_1(X_B)$ and the RRS of second designer is given by $(X_B)_{RRS} = f_2(X_A)$. In order to calculate the RRS explicitly, a designer assumes the set of values for design variables not within their control and chooses values of his/her own design variables in order to maximize his/her own payoff. Since the evaluation of RRS is a computationally expensive process, the function is evaluated at discrete points and a response surface model (or similar approximation technique) is employed to derive an explicit functional form of the RRSs. This process is prone to approximation errors that can be attributed to poor fidelity and low-order functional fit.

The *Nash Non-Cooperative* solution to the coupled, decentralized decision-making problem is the point of intersection of the RRSs pertaining to the different designers. The resulting Nash equilibrium to the design problem has the characteristic that *no designer can improve unilaterally his/her objective function(Thompson 1953)*. The Nash equilibrium thus ensures that each decision-maker's strategy constitutes an optimal response to other decision-makers' strategies. The approach commonly adopted for solving *Nash Non-Cooperative* decision-making problems is explicitly calculating the various RRSs and then finding their intersection. This method represents the use of game theory as a solution algorithm, rather than a communications protocol. Hence, this solution method does not reflect the actual manner in which decisions are made by designers in a decentralized design process. Another solution technique for solving *Nash Non-Cooperative* design problems involves making decisions in an iterative fashion where one designer starts with assumed values for other designers' design variables and makes a decision about his/her own design variables. Other designers use these values in an iterative fashion and determine the values for their design variables under their control. The process continues until the solution converges to the Nash Equilibrium. Although this solution approach more closely resembles interactions associated with decentralized decision-making, convergence and stability are not guaranteed. In order to overcome these respective shortcomings, we offer an alternative game theoretic mechanism for non-cooperative conflict resolution in Section 6.4.

6.3.2 Box Consistency

Box Consistency is a concept stemming from interval arithmetic that is focused on checking the consistency of each equation in a system in order to eliminate sub-boxes of a given box that cannot contain the solution to the system (Hansen and Walster 2004). We implement this construct to successively eliminate those areas of a given design space that do not contain the Nash Equilibrium of the system. Box Consistency constitutes a systematic means of reducing a shared design space that lends itself to turn-based decision making. Since Box Consistency also allows us to embody the propagation of ranged sets of specifications among interacting stakeholders, it is quite suitable as a solution algorithm for coupled, decentralized multifunctional decision-making. Mathematically, Box Consistency can be defined as illustrated at the hand of the following example.

Consider an equation of the form $f(x,y)=0$ such that $x \in X$ and $y \in Y$, where X and Y are intervals. The values of x and y are *consistent* relative to the function f, if for all values of x in X, there exists a y in interval Y, and for all values of y in Y, there exists x in interval X, such that the equation $f(x,y)=0$ is satisfied (see Reference (Hansen and Walster 2004) for a more detailed explanation). This statement can be mathematically represented (where symbols retain their mathematical meaning) as:

$$\forall x \in X, \exists y \in Y \text{ and } \forall y \in Y, \exists x \in X$$

The notion of consistency when extended to higher dimensional spaces translates to Box Consistency. This consistency principle is illustrating in Figure 6-4 using two straight lines, $f_1(x, y) = 0$ and $f_2(x, y) = 0$. The values of $x \in X$ are consistent with values of $y \in Y$ with respect to function f_1 in the figure. Similarly, values of $x \in X$ are consistent with values of $y \in Y'$.

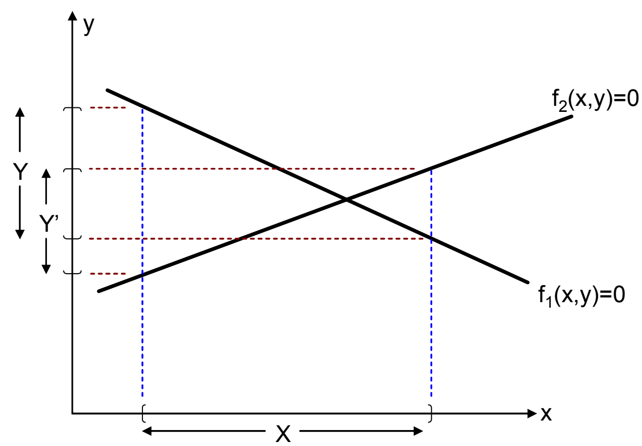


Figure 6-4 - Illustration of consistency

It is important to note that if a box represented by the intervals X and Y is the solution to the set of equations $f_1(x, y) = 0$ and $f_2(x, y) = 0$, then the values $x \in X$ and $y \in Y$ must be box-consistent with respect to both functions f_1 and f_2. For the set of linear equations shown in Figure 6-4, the interval that is box consistent with respect to both functions is thus a single point, specifically the intersection of the two lines making up the system. The same idea is applicable not just for linear functions but for any type of nonlinear function. In order to find the box that is consistent with both f_1 and f_2, a sufficiently large box is chosen and its size is reduced systematically by considering one function at a time until Box Consistency is achieved for all of the functions considered. Assuming that for a subset X_s within the interval X, there are no corresponding values in the interval Y that satisfy the consistency condition, the subset X_s can be excluded because it does not contain the solution. This strategy of systematic reduction of box size is embodied in the interval-based focalization method for decentralized decision-making presented in this chapter and forms the basis for the associated systematic reduction of design freedom. In the next section, we proceed to outline the proposed method and illustrate its application for simple cases.

6.4 An Interval-Based Focalization Method for Decentralized Multifunctional Design

Consider a design problem which is characterized by a set of responses that are associated with different domains. These responses are functions of a common set of design variables, control over which is shared among interacting designers. Hence, coupling is induced, satisfaction of designer objectives is interlinked, and a means of conflict resolution is required. In such scenarios, required interactions among designers are often modeled using principles taken from non-cooperative game theory. Two such approaches, adopted for executing coupled decisions within the literature, center on the explicit calculation of RRS intersections and iterative turn-based resolution as explored by Chanron and co-authors (Chanron and Lewis 2003; Chanron and Lewis 2004; Chanron, Singh et al. 2004).

Both approaches are based on the assumption that different designers control subsets of a common set of design variables and are responsible for satisfying different (and often conflicting) objectives. The first approach involves finding the Nash equilibrium by solving the resulting system of RRS equations explicitly relying on either analytical or numerical techniques. A primary disadvantage of this approach is the computational intensity of RRS evaluation. The second approach centers on iteratively searching the design space for a mutually acceptable solution. Disadvantages of this approach are that iterations may not converge to the equilibrium point and resulting solutions are very sensitive to the initial values chosen for design variables.

In light of these considerations, we propose an alternative *interval-based focalization* method where designer communications are based on ranges of design variables rather than point values. The designers start with a design space, defined by ranges for each design variable as specified by the domain experts, assigned control over them. The interacting decision-makers subsequently proceed to take turns in making decisions about their respective decision variables and progressively reduce the intervals in a systematic fashion until either a sufficient degree of convergence is achieved or all design objectives can be satisfied successfully. This method differs from sequential methods in so far that entire ranges of values (rather than point values) are considered in any given cycle, offering a distinct advantage with regard to changes in objectives and design considerations.

To illustrate this point, assume that N designers are involved in a multifunctional design problem, sharing a common design space defined by a set of design variables v_{ij} (where i ranges from 1 to N, j ranges from 1 to m, and m is the number of design variables controlled by a single designer). This scenario is illustrated at the hand of Figure 6-5. The circle at the center represents the number of design variables. These design variables are partitioned into mutually exclusive sets V_i. For example, Designer 1 has control over sets of design variables $V_1 = \{v_{11}, v_{12}, \ldots, v_{1m}\}$. The arrows in the figure represent the passing of intervals of design variables throughout the design process. As shown in the figure, designers make decisions about their design variables in a cyclic fashion. A designer is in the *active* state if it is his/her turn to make a decision. All other designers are in the *inactive* state. At a given point in time, only one designer is in the active state, while all remaining designers passively observe. A full cycle of the proposed interval-based focalization method is completed once all of the interacting designers have made a decision, successively reducing the available design freedom. The steps of the proposed method are listed in Figure 6-6.

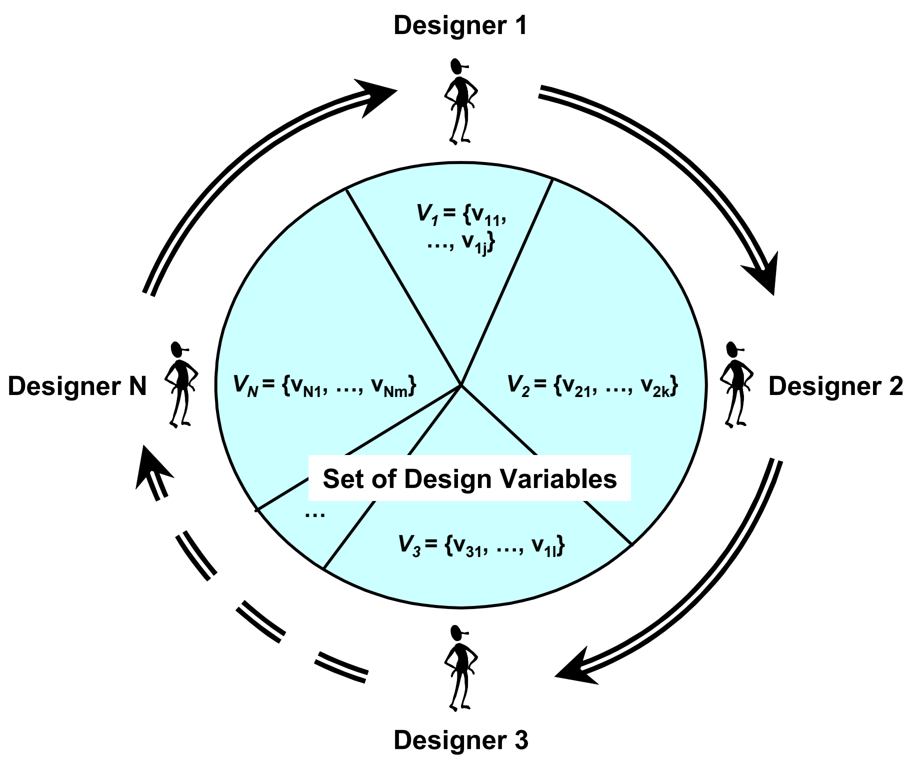


Figure 6-5 - Illustration of interval-based strategy for non-cooperative game theory

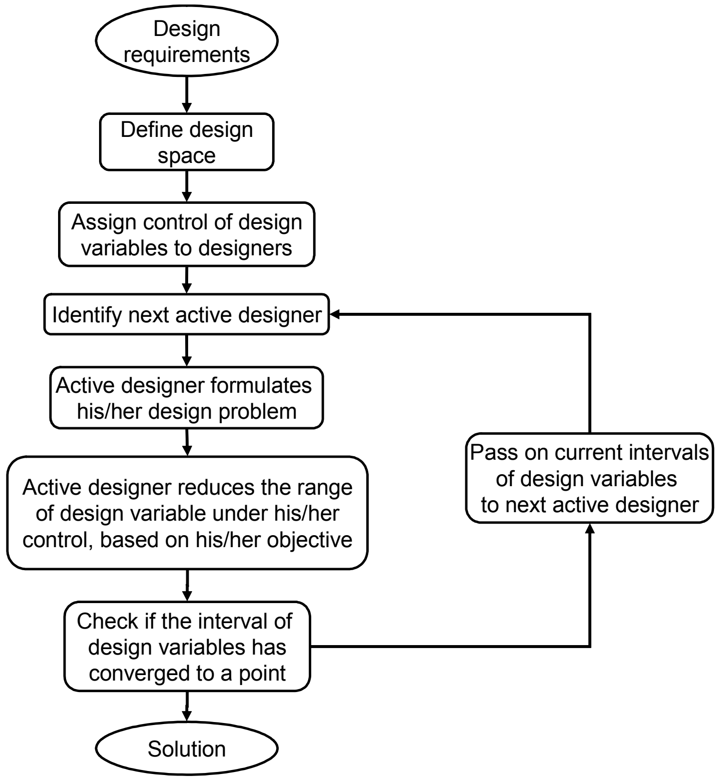


Figure 6-6 – Steps of the proposed interval-based focalization method for decision-making in multifunctional design scenarios

In the proposed method, we represent designer considerations in terms of compromise Decision Support Problems (DSP), the word formulation for which is provided in Table 6-2. During his/her turn, the active designer is presented with a range of values for the design variables controlled by other (inactive) designers. This range represents a set of values within which inactive designers have the freedom (and responsibility) to select any value they choose. This freedom is represented by a double-line arrow in Figure 6-5. Given this range, the active designer determines the largest possible range for his/her design variables that will satisfy his/her response, regardless of what values other designers determine for the decision variables under their control. In other words, the active designer identifies the range of his/her design variables that will ensure Box Consistency with respect to his/her RRS. This is achieved by Newton's Interval Method of Elimination of Intervals that do not satisfy Box Consistency, documented in Ref.

(Hernández, Seepersad et al. 2002). This method requires the identification of lower and upper bounds on unwanted intervals.

Table 6-2 – The compromise DSP word formulation of the decision made by each designer in the interval-based method

Given
Design Problem
Ranges of values for design variables controlled by other designers
Designer's own objective function
Find
Range of values for design variables controlled by active designer
Satisfy
Active designer's design constraints
Lower and upper bounds on design variables
Target values for goals
Minimize
Deviation of active designer's goals from targets

The ranges of values of design variables from the active designer are passed on in sequence to inactive designer who then becomes active during his/her turn. The process is repeated in a cyclical fashion until a sufficient degree of convergence is achieved or all design objectives can be successfully satisfied. Often this degree of convergence is embodied in a single point.

Having described the method, we proceed to illustrate it at the hand of two multifunctional design problems – (1) a scenario with two designers where each designer aims to optimize their responses (see Section 6.4.1) and (2) a scenario where each designer aims to achieve target values for their responses (see Section 6.4.3). A discussion of effects of initial conditions is presented in Section 6.4.2.

6.4.1 Illustrative Example with Linear Rational Reaction Sets (RRS)

In Figure 6-2, we present a scenario where two designers (A and B) are responsible for optimizing responses Y_A and Y_B respectively. Designer A is assigned design variable X_A, whereas designer B controls X_B. In the first cycle, designer A is provided with the range of values for X_B within which, the Designer B has the freedom to select any value. Based on this range of X_B, Designer A determines a range of values for X_A such that for any value of X_B within the specified range, a value for X_A can be chosen that will satisfy his/her response (Y_A). Given this range of X_A, Designer B makes a decision about the range of X_B that will satisfy his/her response (Y_B). Given this range, Designer A revisits his/her decision and the process continues until the values of design variables converge to a point in the design space.

In order to illustrate this method, we focus on a problem with two variables X_A, X_B and two responses Y_A, Y_B. The allowable ranges for design variables are $X_A = [0,10]$ and $X_B = [0,10]$. The responses are related to the design variables as follows:

$$Y_A = \frac{5}{2}X_A^{2} + X_A X_B - 30X_A + \frac{1}{10}X_B^{2} - 5X_B \quad \text{(Designer A)}$$
$$Y_B = -5X_B^{2} - X_A X_B + 50X_B - \frac{1}{16}X_A^{2} + 5X_A \quad \text{(Designer B)}$$

The surface plots for these functions are shown in Figure 6-7. Designer A is responsible for minimizing his/her objective given by the response Y_A, whereas Designer

B is charged with maximizing Y_B. Given that the control is as described, the designers' best response is given by the following equations, that also represent the designers' respective RRSs:

$$X_A = \frac{30 - X_B}{5} \qquad \text{(Designer A's RRS)}$$
$$X_B = \frac{50 - X_A}{10} \qquad \text{(Designer B's RRS)}$$

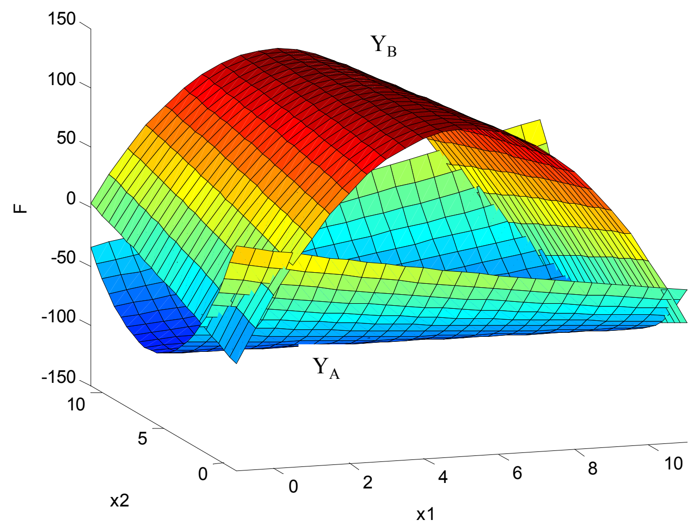


Figure 6-7 - Surface plot for Y_A and Y_B

These RRSs are shown graphically in Figure 6-8. Although this set of linear equations is quite simple, it is nonetheless useful for demonstrating the proposed method. The lines are plotted separately to emphasize that each of the two designers is only aware of his/her responses. The problem is thus representative of a distributed design scenario, where different domain experts are accountable for different (and often conflicting) objectives.

The arrows near the axes indicate designer control with respect to the variable plotted on that axis.

The starting ranges for the two design variables are $X_A = [0,10]$ and $X_B = [0,10]$. Considering the range of $X_B = [0,10]$, Designer A determines the range of X_A that minimizes his/her objective Y_A. This range is evaluated to be $X_A = [4,6]$ using Newton's interval method of elimination of intervals that do not satisfy Box Consistency (Hansen and Walster 2004). It is also clear from Figure 6-8a that all values of $X_A < 4$ and $X_A > 6$ can be excluded from the initial range of X_A, because these values do not lead to Box Consistency with respect to $X_A = \frac{30 - X_B}{5}$. Using the range determined by Designer A, Designer B is able to eliminate those values of X_B from his/her starting range that do not result in Box Consistency with respect to $X_B = \frac{50 - X_A}{10}$. The resulting range for X_B is $X_B = [4.4, 4.6]$. This concludes the first cycle in the interval-based design process. The design spaces resulting from subsequent reductions in the ranges considered by Designers A and B are shown in Figure 6-9a and 7b respectively.

The sequential range reduction cycles continue until the ranges of X_A and X_B converge to a point. The ranges of design variables after successive cycles are provided in Table 6-3. The solution converges to $X_A = 5.103$ and $X_B = 4.489$, a result one might expect based upon the intersection of the designers' respective RRSs.

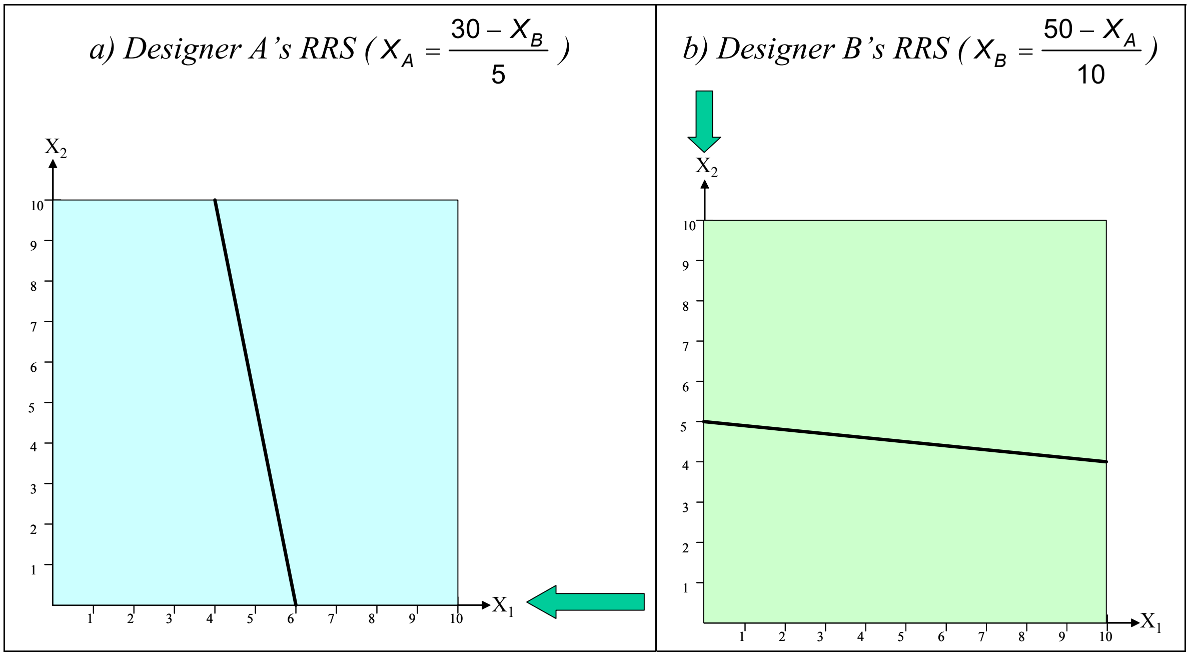


Figure 6-8 - BRCs for Designers A and B

Table 6-3 - Reduction of design range along set-based cycles

Cycle #	***Range for*** X_A	***Range for*** X_B
0	[0, 10]	[0, 10]
1	[4, 6]	[4.4, 4.6]
2	[5.08, 5.12]	[4.488, 4.492]
3	[5.101, 5.102]	[4.489, 4.490]

6.4.2 The Effect of Initial Conditions

A prerequisite initial condition for application of this method is that the starting ranges for variables controlled by both designs are such that it is possible for the active designer to find a value for his/her design variables (satisfying his/her objectives) for all values of design variables controlled by inactive designers. For example in the two-designer scenario, for any value in the range of X_B, Designer A should be able to select a value of X_A that satisfies Y_A. Similarly, for any value in the range of X_A, Designer B should be able to select a value of X_B that satisfies Y_B.

Two important characteristics of this method are –

1 The design space considered in cycle (*i+1*) is always smaller than or equal to the design space in cycle (*i*).

2 If the initial condition is satisfied, all future cycles will also satisfy this condition.

The key advantages of this method are *a)* if the initial condition (mentioned above) is met, the process will never diverge and *b)* there is a gradual reduction of the design space along the process. This means that there is a range of responses that can be satisfied after any given cycle. Hence, if there are changes in design objectives, these can be accommodated without re-executing the design process in its entirety.

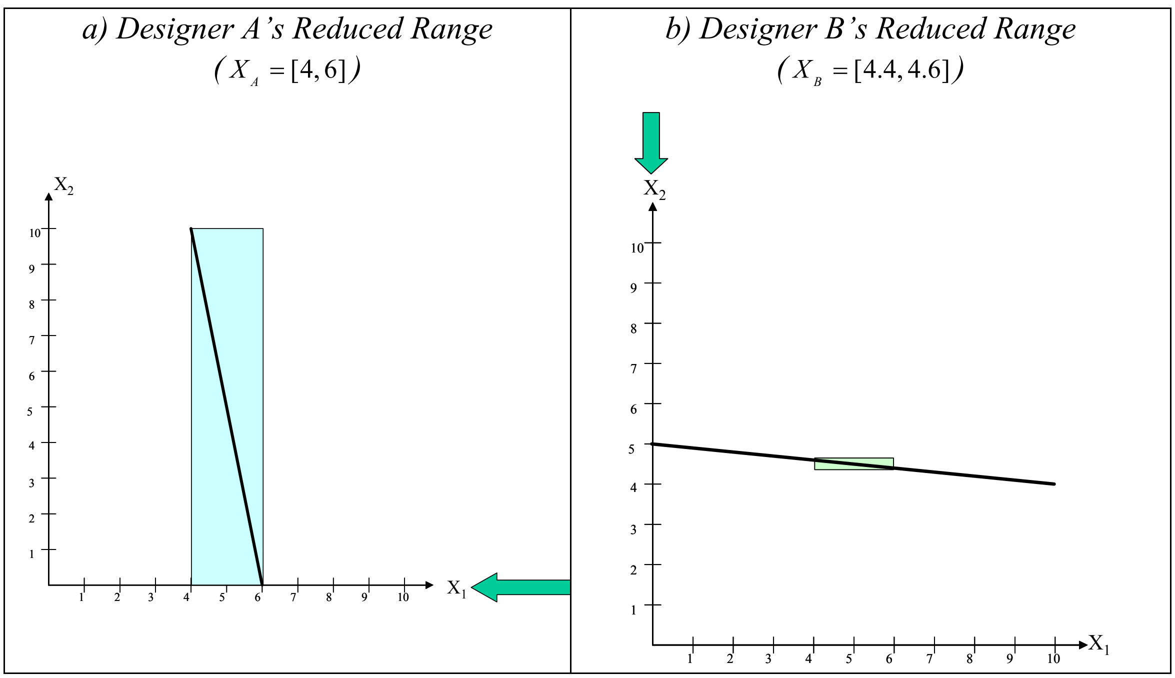


Figure 6-9 - Design space after cycle 1

6.4.3 Illustrative Example for Objective Target Matching – Linear Cellular Alloy Design

In this section, we focus on an example centered on multi-functional design of Linear Cellular Alloys (LCA) (Cochran, Lee et al. 2000; Hayes, Wang et al. 2001) in order to demonstrate the applicability of the proposed interval-based focalization method for complex non-linear problems, where designers aim to achieve target values of their respective objectives. This stands in marked contrast to the example presented in Section 6.4.1, where each designer was interested in maximizing/minimizing their objectives.

Linear Cellular Alloys are honeycomb materials (see Figure 6-10) that are processed via extrusion of ceramic slurry through a multistage die. The slurry is composed of a binder mixed with metal oxide powders. The structure resulting from the extrusion is first dried and reduced into the metallic phase in a hydrogen rich environment. It is then sintered to produce nearly fully dense metal composites. A wide range of cell sizes and shapes including functionally graded structures can be achieved using this manufacturing process. The resulting materials are especially suitable for multi-functional applications that require both strength and heat transfer capabilities (Seepersad, Dempsey et al. 2002). Applications of these materials include heat sinks for microprocessors and combustor liners for aircraft engines. One of the main advantages of these LCAs is that desired structural and thermal properties can be obtained by designing the cell shape, cell arrangement, and cell wall thicknesses, as well as, dimensioning the overall LCA structure.

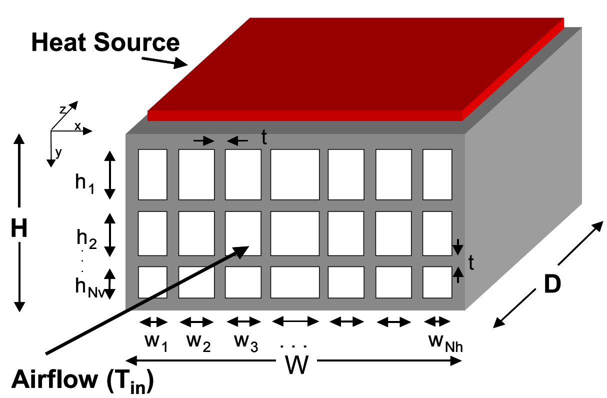


Figure 6-10 - Linear Cellular Alloy with rectangular cells

Consider a scenario where a multi-functional LCA is to be designed with the following behavioral attributes –

- Overall heat transfer rate ($\dot{Q}$) = -5.6183 W
- Compliance (C) = 4.4773 kJ

The design problem involves evaluation of design variables values. In this case, the following two geometric parameters of the LCA can be varied – overall height of LCA (H), and wall thickness (t). All other parameters in the LCA geometry are fixed. Designer A controls overall height (H) and is responsible for achieving the targeted total heat transfer rate ($\dot{Q}$). Designer B is responsible for Compliance and controls the wall thickness of the rectangular LCA. The design variables, responses, and their associated control are shown in Figure 6-11.

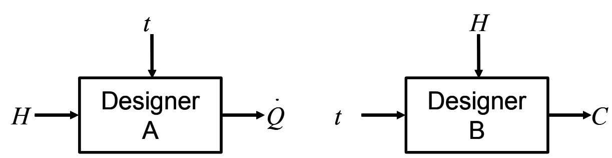


Figure 6-11 - Control of design variables in LCA design scenario

The results obtained by applying the proposed interval-based focalization method (see Figure 6-6) are presented in Table 6-4. In this table, the ranges of design variables (overall height and wall thickness) after each successive cycle are presented. The gradual reduction of the design space along the design process is plotted in Figure 6-12. This example shows that the proposed method can be applied to non-linear problems as well. It is important to note that after each cycle, the achievable target values for compliance and overall heat transfer are also ranges. This example demonstrates that the proposed interval-based focalization method can also be applied successfully to non-linear problems.

Table 6-4 - Ranges of design variables at different cycles

Cycle #	*Range for* thickness (t) (mm)	*Range for* Height (H) (mm)	*Range of Achievable Heat Transfer Rates* (W)	*Range of Achievable Compliance* (kJ)
0	[0.0045, 0.0065]	[10, 30]	[-6.713, -4.699]	[3.87, 5.62]
1	[0.005452, 0.006298]	[15.33, 21.64]	[-6.028, -5.236]	[4.18, 4.85]
2	[0.005789, 0.006091]	[17.47, 19.72]	[-5.760, -5.479]	[4.37, 4.60]
3	[0.005905, 0.006012]	[18.28, 19.09]	[-5.668, -5.568]	[4.43, 4.52]
4	[0.005945, 0.005983]	[18.581, 18.87]	[-5.635, -5.600]	[4.46, 4.49]
5	[0.005959, 0.005973]	[18.68, 18.79]	[-5.624, -5.612]	[4.47, 4.48]
6	[0.005964, 0.005969]	[18.72, 18.76]	[-5.620, -5.616]	[4.47, 4.47]
7	[0.005966, 0.005968]	[18.74, 18.75]	[-5.619, -5.617]	[4.47, 4.47]
8	[0.005967, 0.005967]	[18.74, 18.75]	[-5.618, -5.618]	[4.47, 4.47]
9	[0.005967, 0.005967]	[18.74, 18.74]	[-5.618, -5.618]	[4.47, 4.47]

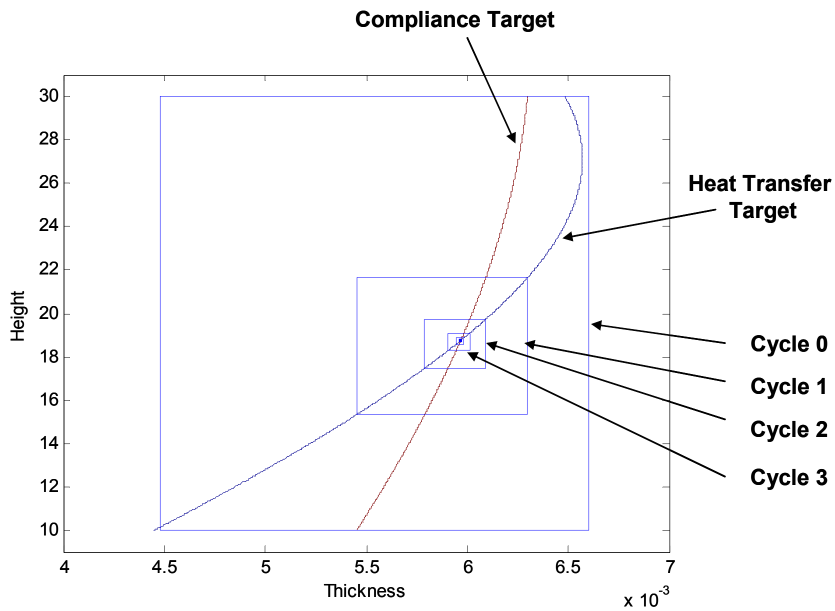


Figure 6-12 - Convergence of decisions to Nash Equilibrium

6.4.4 A Convergence Criterion for the Interval-Based Focalization Method

An important criterion for successful application of a turn-based method is its convergence characteristic. It is on this aspect that we focus in this section. For more *general design scenarios* where the RRSs are not necessarily linear, a similar convergence criterion applies. If the RRSs for Designers A and B take the functional forms f_A and f_B, respectively, they can be represented mathematically as:

$$X_A = f_A(X_B)$$
$$X_B = f_B(X_A)$$

In this case, the criteria governing convergence are the following - successive intervals of each design variable must be proper subsets of intervals determined during previous cycles. This is mathematically represented as:

$$[X_A]_{i+1} = f_A([X_B]_i) \subset [X_A]_i$$
$$[X_B]_{i+1} = f_B([X_A]_{i+1}) \subset [X_B]_i$$

The following discussion draws on the example presented in Section 6.4, where Designers A and B control X_A and X_B, respectively. In order to illustrate the impact of design variable control on process convergence we assume that the control of design variables is reversed (i.e., the Designer A now controls variable X_B and Designer B has control over X_A). The objectives and starting ranges for variables remain the same, however. Changing the control over design variables results in a different set of RRSs. These RRSs are mathematically given by the following expressions:

$$X_B = 25 - 5X_A \qquad \text{(Designer A's RRS)}$$

$$X_A = -8X_B + 40 \qquad \text{(Designer B's RRS)}$$

The RRSs for this case are shown in Figure 6-13. The arrows represent control over design variables.

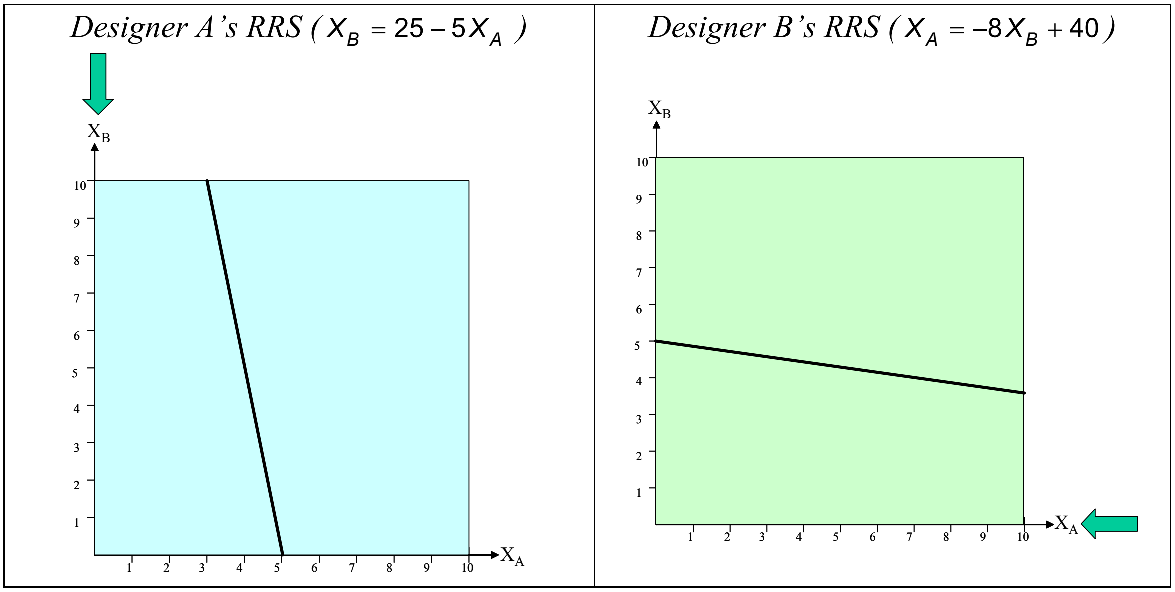


Figure 6-13 - BRCs for Designers A and B when A controls X_B and B controls X_A

In the first cycle, Designer A determines the range of X_B corresponding to the starting range of $X_A = [0,10]$ (controlled by Designer B) such that his/her objectives are satisfied. Given this range for X_A, the required range for X_B is $X_B = [0,10]$ as shown in Figure 6-14. Using this range for X_B, Designer B determines the required range for X_A (the variable under his/her control) to be $X_A = [0,10]$. Continuing this process does not result in convergent behavior, underscoring the fact that the *assignment of control over design variables to different designers indeed has an effect on the convergence of the underlying process*. However, in contrast with the point-based methods discussed in Section 6.2, the designers are able to identify in a single cycle whether the solution is going to converge. In a point-based method, divergence would not have been obvious and continued iteration would have been required. The benefit of developing convergence criteria for interval based methods is that the criteria serve as a guide for appropriate partitioning of the design variable set into subsets assigned to designers.

Based on this simple example, it becomes apparent that there is a need to develop a criterion for convergence of the interval-based focalization method. The method will converge if the range of design variables at cycle *(i+1)* is a subset of the range of design variables at cycle *i*. In other words, when the design space is effectively reduced after each cycle. The notation used for representing ranges of design variable X after cycle *i* in this section is - $[X]_i$, where $[X] = [X_{min}, X_{max}]$.

Based on the RRSs for the scenario presented in Section 6.4, the convergence criteria is that the range of variables X_A and X_B during cycle *(i+1)* should be less than the corresponding ranges during cycle *i*. This is represented mathematically as follows –

$$[X_A]_{i+1} = \frac{30 - [X_B]_i}{5} \subset [X_A]_i$$

$$[X_B]_{i+1} = \frac{50 - [X_A]_{i+1}}{10} \subset [X_B]_i$$

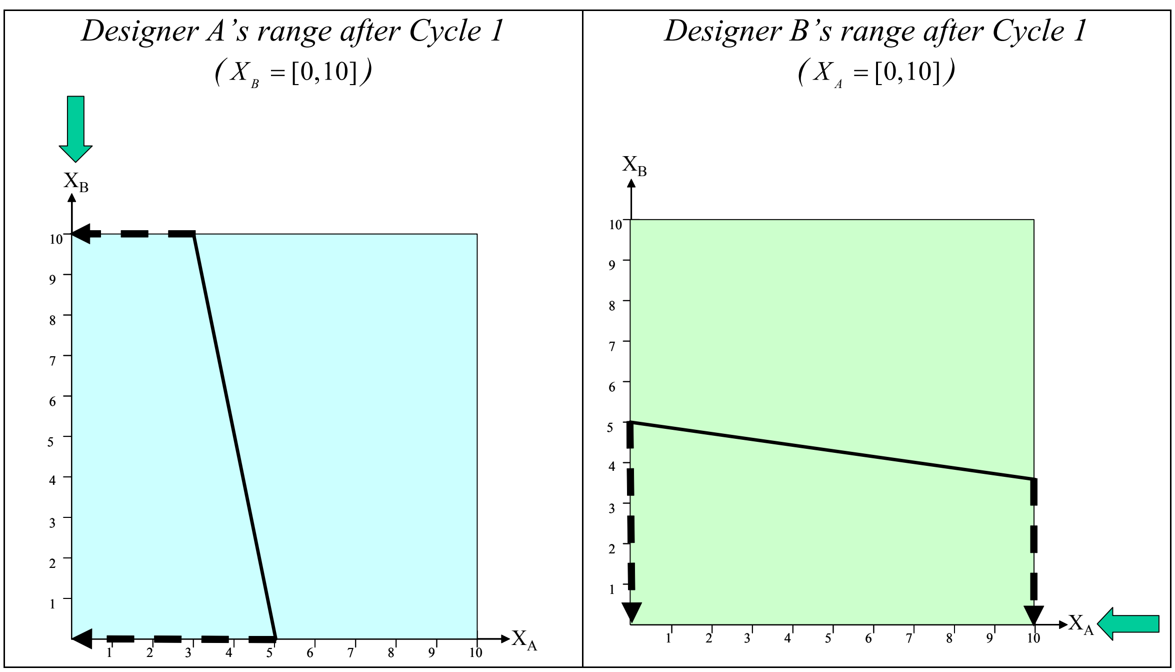


Figure 6-14 - Designers' ranges after Cycle 1

Using the values of Y_A and Y_B, these convergence criteria can be evaluated to $X_{B,\min} < 4.489 < X_{B,\max}$ and $X_{A,\min} < 5.103 < X_{A,\max}$. This effectively means that the starting ranges of X_A and X_B do not affect convergence of the solution. This is in contrast to the point-based iterative method of Ref. (Chanron and Lewis 2003), where the choice of starting points directly affects convergence.

In the scenario where design variable control is reversed, the convergence criteria are:

$$[X_A]_{i+1} = -8[X_B]_i + 40 \subset [X_A]_i$$
$$[X_B]_{i+1} = 25 - 5[X_A]_{i+1} \subset [X_B]_i$$

Evaluating these expressions shows that $X_{B,\min} > 4.4872 > X_{B,\max}$ and $X_{A,\min} > 4.1026 > X_{A,\max}$. Obviously, this is not possible, since the minima in each range exceed the maxima. Based on the convergence criterion, it is clear that this design process will not and, in fact, *cannot* converge. This underscores that different partitioning schemes may not only lead to different answers but may also change the convergence characteristics underlying a design problem. We thus assert that *the proposed convergence criterion can be used as a basis for the assignment of design variable control to different designers*. This is the next issue we plan to explore in future in developing the proposed interval-based focalization method further.

6.5 On Verification and Validation

An overview of the validation of interval-based focalization method for multifunctional design presented in this chapter is provided in Figure 6-15. This validation square provides the validation details specific to multifunctional design aspects of overall dissertation level validation square presented in Figure 1-13. In this chapter, three quadrants of verification and validation are addressed – theoretical structural validation, empirical structural validation, and empirical performance validation. These quadrants of validation square are discussed in Sections 6.5.1, 6.5.2, and 6.5.3 respectively.

Theoretical Structural Validity

Validity of the constructs of the method

- ***Chapter 6*** **- Literature review on decentralized multifunctional design**
  - ✓ **Review of existing literature on *a)* Game-based protocols and *b)* Interval based constructs (box consistency)**
  - ✓ **Relevance of these two constructs in multifunctional design problems**

Theoretical Performance Validity

Usefulness of the method beyond examples

- ***Chapter 10*** **- Arguing the validity of framework developed in this dissertation beyond the examples used**

Empirical Structural Validity

Appropriateness of the examples chosen to verify the method

- ***Chapter 6*** **- Multifunctional LCA design**
  - ✓ **Simple example for illustration purposes – quadratic response surfaces, resulting in linear RRSs, direct calculation of convergence rate**
  - ✓ **LCA design problem for validation – response not available in explicit form, non-linear RRSs**

Empirical Performance Validity

Usefulness of the method in examples

- ***Chapter 6*** **- Functional Decoupling using LCA design**
  - ✓ **Application of the method to two problems with different characteristics and complexity resulted in convergence to Nash equilibrium**
  - ✓ **The set-based focalization method demonstrated to be better in linear RSS case due to non-divergence**

Figure 6-15 - Validation of the method for decentralized, multifunctional design presented in Chapter 6

6.5.1 Theoretical Structural Validation

Theoretical structural validation refers to accepting the individual constructs constituting the method and accepting the internal consistency of the way the constructs are put together. Theoretical structural validation is carried out in this chapter using a systematic procedure consisting of *a)* identifying the method's scope of application, *b)* reviewing the relevant literature and identifying the strengths and limitations of constructs available in the literature, and *c)* identifying the gaps in existing literature and determining which constructs can be leveraged in the design method.

The theoretical constructs used in this chapter include game theory based protocols and interval based constructs such as box consistency. Game theory based protocols are used previously in the design literature. A review of existing literature on game theory for decentralized design is provided in Section 6.2. Based on this literature review, it was identified that the current methods have limitations related to convergence, and the manner in which design freedom is reduced. In order to address these limitations, integration of set-based design and game theory based protocols is proposed because that would help help designers in communicating sets of possible solutions, thereby keeping design freedom for a longer period of time. The advantages of both these constructs have been shown independently in the design literature, which gives us confidence in the applicability of individual constructs. Although the integration of these two constructs is not performed in the existing literature, their internal structure is not incompatible. The integration of both these constructs in a manner that combines advantages of both is performed in this chapter.

6.5.2 Empirical Structural Validation

Empirical structural validation refers to accepting the appropriateness of example problems used to verify the performance of the method. In this chapter, we use two example scenarios for validation of the interval-based focalization method. These two examples are different in complexity and in the manner in which response is evaluated based on the design variable values. In the first example, the response is given as simple quadratic equation in terms of the design variables. This example is chosen because it results in a linear set of rational reaction sets, which is easy to evaluate and has a constant rate of convergence. Further it allows easier explicit calculation of convergence criterion and allows the study of initial condition.

The second example (LCA design) is more complex. In this case, the response is not provided explicitly as polynomial equations. The response is evaluated using MATLAB based thermal and structural behavioral models. The rational reaction sets in this case are non linear. The example is used to check if the method illustrated using linear rational reaction sets also works for non-linear cases. Hence, we believe that these two examples are appropriate for demonstrating the validity of the interval based focalization method developed in this chapter.

6.5.3 Empirical Performance Validation

Empirical performance validation refers to accepting that the outcome of the method is useful with respect to the initial purpose for some chosen example problems and accepting that the achieved usefulness is linked to applying the method. It is shown in Sections 6.4.1 and 6.4.3 that the method applied to both examples results in convergence of the solution to Nash equilibrium. This demonstrates that the method produces valid results. A case where the method does not converge is shown in Section 6.4.4. However, in this case, the range of values for the design variables does not reduce after the first cycle. Hence, in a single iteration, the designers can understand that a particular assignment of design variables does not converge, without explicitly evaluating the convergence criterion (which may be difficult in cases where rational reaction sets are not known in explicit form). If the same example is used with the point based iterative method, the solution would diverge and there won't be any way of assessing whether the solution is converging converge or not (of course, without calculating the convergence criterion explicitly). This advantage is primarily due to the application of set based focalization method presented in this chapter. Hence, we can say that empirical performance validity is achieved.

6.5.4 Critical Review and Limitations of the Proposed Method

In this chapter, an interval-based focalization method is presented for facilitating interactions, modeled using non-cooperative game theoretic protocols, as commonly employed for conflict resolution in decentralized, multifunctional design scenarios, involving shared control over design variables. This method is based on the Box Consistency principle, developed in the area of interval arithmetic. Key advantages of adopting the proposed method include non-divergence of solutions to coupled design problems, insensitivity of convergence characteristics to starting ranges, and gradual reduction of design freedom, prolonging adaptability to design changes. The proposed method is illustrated at the hand of two examples. Specifically, we solve a non-cooperative game, centered on the intersection of linear RRSs, in the first example (a set of quadratic equations) and underscore the influence of control assignment on convergence. Application of the method to cases where RRSs are non-linear is demonstrated in the second example (LCA design).

Further development of this method is centered on investigating cases where – *a)* designers have additional, local design variables that are not shared, but depend on the values of shared parameters, *b)* design variables are defined on discontinuous or piece-wise defined intervals, and *c)* multiple non-cooperative solutions (Nash equilibrium) exist. In the case of multiple Nash equilibria, convergence to a point solution may be impeded. This means that the size of the box may remain constant from one cycle to the next. Consider the case of two designers with RRSs intersecting more than once, as shown in Figure 6-16(a). After several reductions of the design space using the proposed interval-based focalization method, the region containing possible solutions is reduced to rectangle ABCD. Clearly, subsequent cycles will not reduce the design space further. A possible solution to this problem is to partition the design space into subsets (e.g., rectangles AEFD and EBCF in Figure 6-16 (b)) to which the focalization method is then applied in parallel. The result of one post partition cycle is shown in Figure 6-16 (c).

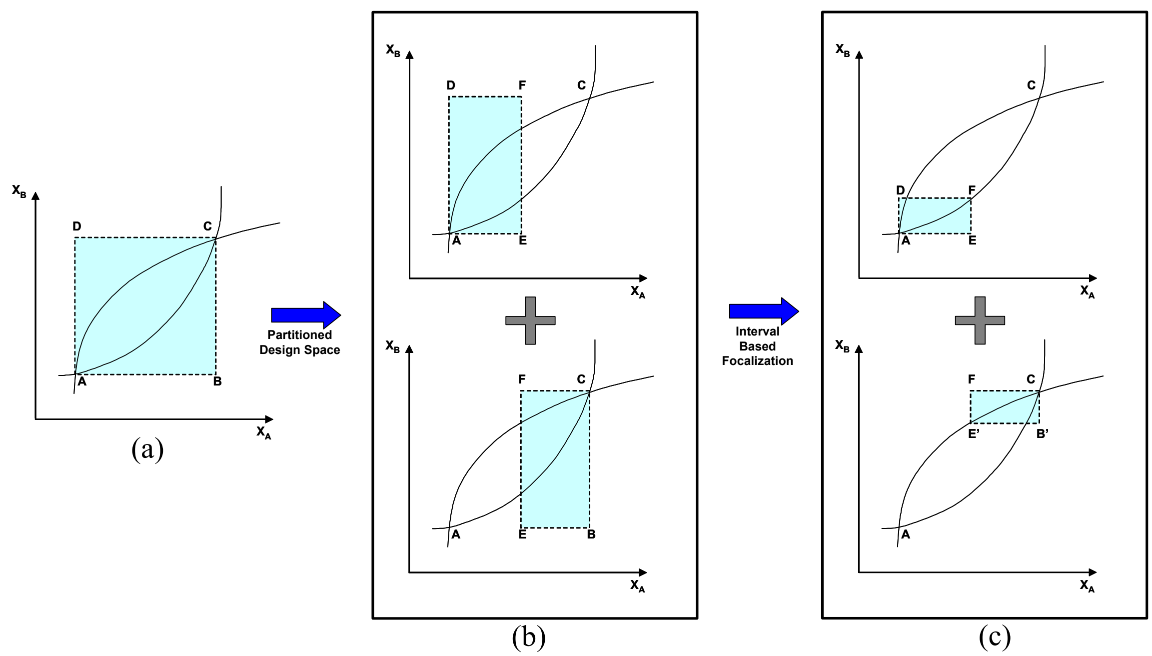


Figure 6-16 - Handling multiple Nash equilibria

6.6 Role of Chapter 6 in This Dissertation

The interval based focalization method presented in this chapter is developed for decentralized, multifunctional design scenarios. This method does not address meta-design, but is an important design process for decisions related to different functional requirements that are *a)* strongly coupled with each other, and *b)* controlled by distributed designers. Hence, this method can be used in Step 4 of the design method presented in Chapter 3. The relationship of this chapter with Chapter 3 is shown in Figure 6-17. The focus in the first six chapters of the dissertation is on methods and metrics for integrated design of products and design processes (RQ 1 and RQ2). We now shift our attention (in Chapter 7 and Chapter 8) to the third research area that relates to modeling and representation of design information that supports integrated design of products and design processes.

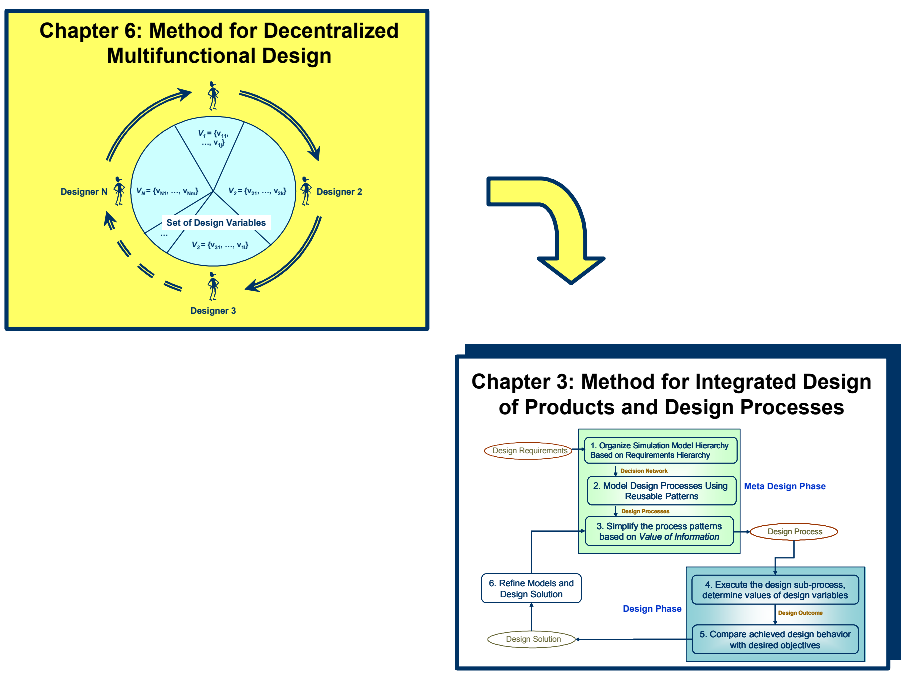
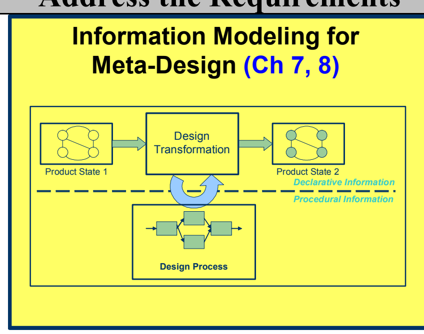
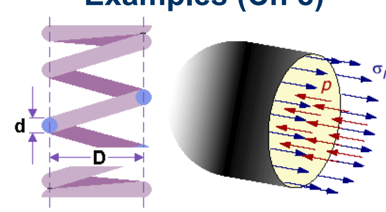


Figure 6-17 – Relationship of Chapter 6 with other chapters in the dissertation

Chapter 7 Modeling Design Processes – A Systems Approach

In this chapter, we address the third research question - "*How should simulation-based design processes be modeled in a systematic manner and represented in a computer interpretable format to support meta-design*". This research question is a result of the sixth requirement presented in Table 1-5. The requirement, associated component of the framework addressed in Chapter 7 and Chapter 8, and the validation examples are presented in Table 7-1.

Table 7-1 – Requirements, framework component, and validation example presented in Chapters 7 and 8

Framework Requirements	Components of the Framework Developed to Address the Requirements	Validation Examples
6) Support design process exploration, and reusability of existing design process, product and decision related information and knowledge		**Pressure Vessel, Spring Examples (Ch 8)**  ***Purpose:*** To demonstrate the approach for supporting meta-design in computational frameworks

As a background on design information modeling, a review of existing literature on modeling design information is presented in Section 2.6. An overview of the issues related to answering this question in the context of simulation-based design frameworks is provided in Section 7.1. Abstracting from the literature review, a set of requirements to be satisfied using the proposed model is listed in Section 7.1.1. The key requirement is to model design information in a manner supporting design process exploration. The proposed strategy for addressing this requirement is presented in Section 7.2. The strategy is based on the hypothesis that – "Separation of product, process, and problem related information enhances reusability of design process information across different products, thereby supporting meta-design". The aspects of research addressed in this chapter are highlighted in Figure 7-1. The hypothesis addressed in this chapter and the relationship of this chapter with other chapters is shown in Section 7.2. The implementation of this strategy in the form of an information model is presented in Chapter 8.

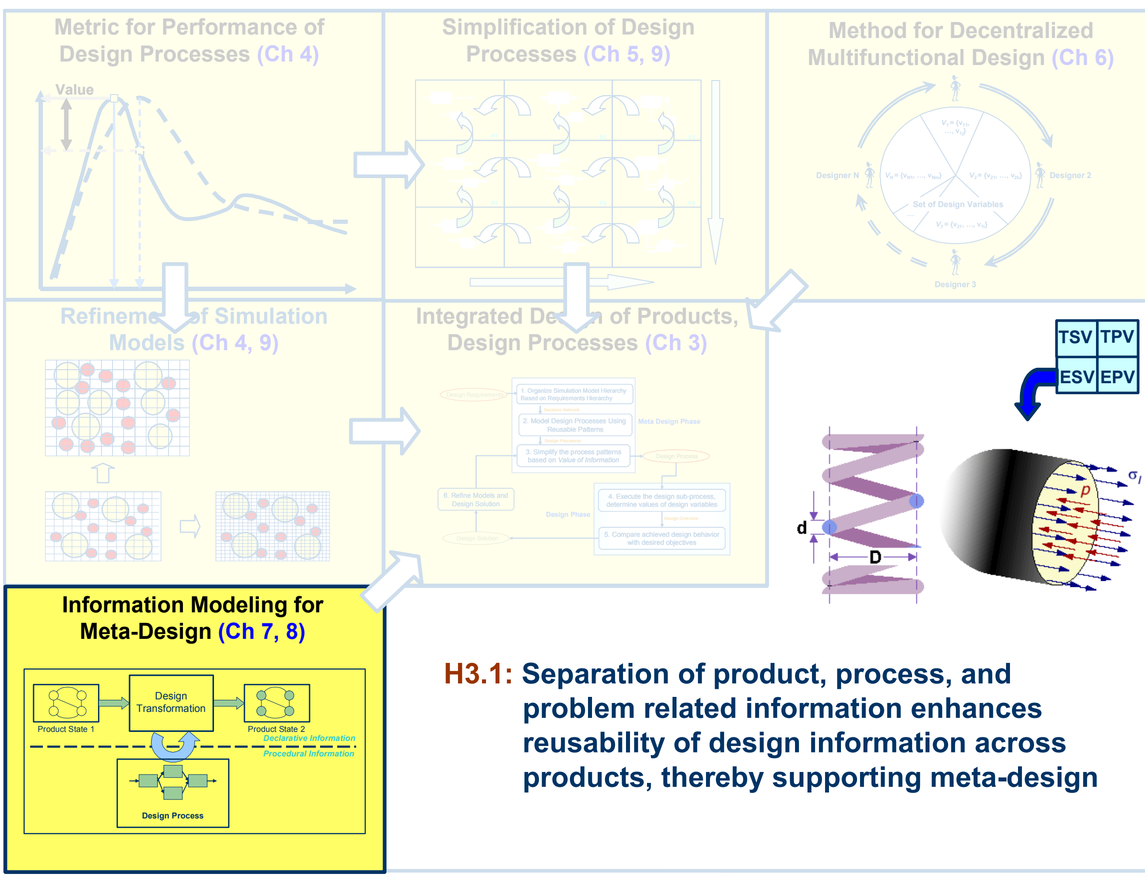


Figure 7-1 –Hypotheses addressed in Chapter 7

7.1 Context: Answering the Research Question 3 – Modeling and Representation of Design Processes

In simulation-based design, the *design process* represents the manner in which information, generated by simulation models, is utilized for satisfying design objectives through analysis, synthesis, and evaluation. These processes are inherently complex because of the interdependencies among simulation models at various scales. Given this inherent complexity of design processes, it is imperative that the *design processes themselves be designed appropriately* and *systematically*. Inefficient design processes can lead to longer design timelines, thereby contributing to higher costs (Bras and Mistree 1991). The methodical design of design processes is also referred to as *meta-design*. The role of meta-design in product design is well acknowledged throughout the dissertation. According to Simon, "… design process strategies can affect not only the efficiency with which resources for designing are used, but also the nature of final design as well" (Simon 1996). Bras and Mistree (Bras and Mistree 1991) point out that "a necessary ingredient in increasing the efficiency and effectiveness of human designers is the modeling of design processes in a manner that can be analyzed, manipulated and implemented". The systematic design of design processes is thus crucial for the timely deployment of products. Panchal and co-authors (Panchal, Fernández et al.) highlight that design processes are a company's primary intellectual capital and should be designed, managed, and reused strategically.

In spite of the fundamental importance of meta-design in expending resources, it is not effectively supported by current Computer Aided Engineering (CAE) and Product Lifecycle Management (PLM) frameworks. The question, naturally arising from this observation is: *"How should CAE and PLM frameworks be developed/modified to support meta-design?"* Although this query can be posed for most design frameworks, we primarily focus on simulation-based design frameworks such as FIPER, ModelCenter and iSIGHT. Such CAE and PLM frameworks adopt a *tool-centric* view of design processes, according to which a design process is a network comprised of software tools employed for processing information. The adoption of a tool centric perspective in developing design frameworks, thus invariably focuses the underlying effort on *achieving interoperability* between 1) different tools that perform similar function (such as different CAD applications), 2) tools providing different functionality (structural analysis, crash, vibration, etc.), and 3) applications pertaining to different domains. Various standards such as STEP, XML, and UML are being developed to achieve interoperability between such tools. Recently, Peak and co-authors (Peak, Lubell et al. 2004) proposed a *model-centric* perspective to support the further development of these frameworks. Specifically, a product information model comprises a central core, modified and populated using all relevant tools. Such a model-centric view constitutes a significant improvement over the tool-centric view, commonly espoused, because information is no longer tied solely to the particular tools used for its creation or modification. We acknowledge that a model-centric perspective is important for realizing the seamless integration of information models, associated with different aspects of product design, and useful for guiding the development of CAE and PLM frameworks to support fine grained interoperability, as well as, the development of a collective product model. However, we assert that neither the tool-centric nor model-centric perspectives (alone or in concert) are adequate for effectively supporting meta-design.

A fundamental obstacle in furnishing the capability for meta-design is the inability of current tools to capture the problem solving aspect of design. In fact, such tools are primarily used to capture procedural aspects. Put another way, current tools do not capture *a) what* the design problem is, *b) how* the designer partitions the problem, and *c) how* different problems are related. Instead, current tools only capture the specific series of steps a designer adopts when solving the problem at hand in a quasi documentary fashion. Design problem changes can thus not be translated to the procedural information captured within the individual tools.

The word "*problem*" has been used in many different ways in the engineering design community. In this dissertation, we define a problem as "either an obstacle to be overcome or a question to be answered". This definition is taken from (Muster and Mistree 1988). This definition is different from the text book type problem solving, where the problem is completely defined and can be solved using a predefined set of steps resulting in a unique solution (see (Hazelrigg 1998)). In real design scenarios, designers are faced with problems where complete information for solving the problem is not available and the closed form solution is not available. Without capturing the problem solving aspect of design in the CAE and PLM frameworks, it is difficult to support meta-design.

We believe that the solution to this problem lies in adopting a decision-centric, problem-solving approach to design. According to many researchers such as Hazelrigg (Hazelrigg 1998), Muster and Mistree (Muster and Mistree 1988), and Thurston (Thurston 1999) the fundamental premise of decision-based design is that engineering design is primarily a decision-making process. A decision-centric approach is adopted in this dissertation because from a decision-centric perspective, meta-design is a meta-level process of designing systems that includes partitioning the system based on function, partitioning the design process into decisions, and planning the sequence in which these decisions are most appropriately made (Mistree, Smith et al. 1990).

Specific advantages of adopting a decision-centric perspective include the ease with which both model-centric and tool-centric views are generated. Furthermore, domain independent representation of design processes becomes feasible. Hazelrigg describes decision-based design as *omni-disciplinary,* "the seed that glues together the heretofore disparate engineering disciplines as well as economics, marketing, business, operations research, probability theory, optimization and others" (Hazelrigg 1998). Herrmann and Schmidt (Herrmann and Schmidt 2002) describe a complete product development organization as a network of decision-makers who use and create information to develop a product. Although principles of decision-based design have been accepted in theoretical aspects of design research, they have not been implemented in design frameworks. Current tools do not capture information related to designers' decisions; decision related information is captured in the form of meta-data (if at all).

To address the shortcomings, elicited throughout this section, and support meta-design in design frameworks, we propose a decision-centric *3-P* approach in this dissertation. The three main elements are *a)* decision-based design (discussed in Section 7.2.1), *b)* modular systems view of design processes (discussed in Section 7.2.2), and *c)* separation of declarative and procedural information (discussed in Section 7.2.3). The utilization of these three elements in the proposed approach is presented in Section 7.3. An information model supporting the *3-P* approach is presented in Chapter 8. It is an object-oriented information model that captures three key components of design information, including *a)* design ***problem***, *b)* design ***process***, and *c)* ***product*** information. The information models for design problem, product, and processes are discussed in Sections 8.1, 8.2, and 8.3 respectively.

7.1.1 Requirements for Modeling Design Information to Support Design Process Exploration

Most of the modeling efforts for design process models are focused on either understanding or capturing the design process for later use. Our focus in this research is computer-supported design of design processes. In order to achieve this goal, the design processes should be modeled in a manner that *supports both analysis and synthesis of processes*. Current design process representations do not lend themselves to the analysis of impact of process on the product. All these existing design process models are useful for investigating "how and when" a task needs to be *performed* and provides little insight into "what" a designer does in an activity (i.e., what is the impact of an activity on the product), which is the focus of this research. The key benefit of modeling a design process is the ability to understand the impact of the process on the product and to configure the process to achieve desired goals. Hence, there is a need to model the design process and the product in a manner that they can be linked together.

Before providing details about the existing efforts in design information modeling, we first discuss the requirements from the point of view of designing design processes. The summary of these requirements is listed in Table 7-2. The requirements are divided into the following categories – A) Support for designing design processes, B) Modeling process information, C) Modeling product information, and D) Reuse of information. In this section, we discuss requirements associated with these categories in detail. It is recognized that there are a lot of other requirements for modeling design information. However, only the requirements that we feel are critical from the perspective of designing design processes are included here.

Table 7-2 – Requirements list for modeling design information

Requirements for design information modeling
A) Support for designing design processes
1. Existence of mathematical models for design processes and products
2. Ability to model linkages between mathematical model and computational model to support execution
3. Support for design decision making (the information model should capture designers' preferences, goals, etc.)
4. Ability to define design problems (that capture the knowledge associated with a particular design transformation)
5. Ability to identify better designs and suitable courses of actions
B) Modeling process information
6. Capability to define processes at all these levels of abstraction
7. Support for Composability of sub-processes into overall processes
8. Separation of problem formulation from process information and tool specific execution details
C) Modeling product information
9. Capability to understand the evolution of product information along the design process
10. Ability to generate meta-information about the design space (such as the size of design space, coupling between parameters, independence, etc.)
11. Ability to representation uncertain information
D) Reuse of information
12. Support reusability of processes at computational level
13. Modular use of processes for different products
14. Modular use of processes for different design problems

A) Support for designing design processes

There is a clear dichotomy of focus in the research for supporting engineering design. On one hand, the focus is on developing mathematically rigorous models for gaining an understanding of the process of design in general. Examples of such efforts include

(Chandrasekaran 1990; Coyne 1990; Gero 1990; Maher 1990; Suh 1990; Takeda, Veerkamp et al. 1990; Simon 1996). On the other hand, design research is focused on developing computer-based methods and tools for providing computer-based support for design. This trend is also evident in modeling design processes. Some of the efforts are focused on developing mathematical models for design processes whereas other efforts are focused on developing information models for design processes that facilitate capturing associated information in a consistent manner in engineering databases. The first requirement in Table 7-2 is the *existence of mathematical models for both products and design processes*. These mathematical models should be compatible with each other. The primary advantage of studying mathematical models of the design processes is to quantify the appropriateness of one design process over the other in a given design scenario, whereas, the advantage of developing information models is to store and reuse information for efficient and fast design. In spite of the complementary nature of both these activities, there has been little work on bridging these two diverse ends of the spectrum. To corroborate this, Zeng and Gu (Zeng and Gu) pointed out that science-based design is still in prehistoric stage due to a lack of a good combination of precise representational languages and laws governing design processes. This brings into light the requirement for bridging these two efforts – *ability to model linkages between mathematical model and computational model to support execution* (see Table 7-2).

Since design is a decision making activity (Mistree, Smith et al.), the mathematical models and the information models should *support for design decision making.* Hence, they should support capturing information such as designers' preferences, goals, etc. Current product and process models do not capture this information. This brings into light the need for capturing *design problems*. Existing design process models are useful for investigating "how and when" a task needs to be performed and provides little insight into "what" a designer does in an activity (i.e., what is the impact of an activity on the product), which is important for designing design processes. Design problems captures to the "what" of activities in a design process. Since information about the design problem is not currently captured explicitly, it is not possible to capture the rationale behind selecting a design process for solving a design problem. This hinders reusability of previously utilized design processes and prevents design of design processes at a computational level. This requirement is listed in Table 7-2 as the *ability to define design problems*.

Our focus in this dissertation is on designing both the products and design processes in an integrated fashion. As shown in the previous chapters, the designers make decisions both about products and design processes. Hence, the modeling approach should support decision making related to products and design processes, which translates to the requirement - *ability to identify better designs (i.e., decisions about the product) and suitable courses of actions (i.e., decisions about the design processes).*

B) Modeling process information

Design processes can be defined at various levels of abstraction – with the business level inter-organizational processes at the higher level and simulation execution towards the lower level of abstraction. Depending on the level of abstraction of design processes, the scope also changes as shown in Figure 7-2. The design process models should have the *capability to define processes at all these levels of abstraction*.

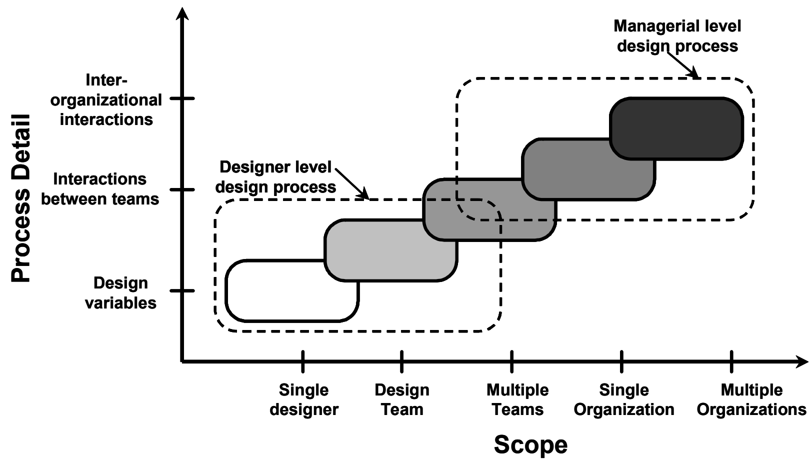


Figure 7-2 - Modeling processes at various levels of abstraction

The approach for designing design processes proposed in this dissertation is to model design processes as modular subsystems that can be composed to develop higher level design processes. This should be supported by the information model. The sub-processes should be modular in nature such that it is possible to *compose design sub-processes into overall design processes* at an executable level.

As discussed before, the information model should also capture the problem related information. Since associated with each design problem, there can be many design processes; to *separation of problem formulation from process* information is necessary. In addition to this, separation of process information with information specific with execution on different tools is also important. This is required to enable utilization of a variety of processes for a problem and also the utilization of various tools for execution of a process.

C) Modeling product information

A design process can be viewed as a network of transformation of information from one state to another. A prerequisite for designing the design process is to understand how product information changes throughout the design process and the impact of each process step on the product information. This is captured as the following requirement in Table 7-2 - *capability to understand the evolution of product information along the design process*. This is possible only if the product information either captures or is able to generate information about the design space such as its size, topology, coupling between the parameters, etc. This is also a part of the knowledge about the design that increases along the design timeline. Hence, the requirement on the product information is - *ability to generate meta-information about the design space*. Further, there is a lot of uncertainty in the design information that must be captured in the product model. This includes uncertainty in the design requirements, the environment, decisions made by other designers, etc. As the design progresses, uncertainty in the product information reduces. Hence, the product model should have the *ability to representation uncertain information.*

D) Reuse of information

In order to perform design process exploration, one of the fundamental issues that need to be addressed is - capturing design related information from past design scenarios and reusing it. This reuse can be done at various levels such as symbolic level, description level, computational level, etc. Since our focus is on simulation-based design, we are interested in reusability at a computational level, and hence listed as the following requirement - *support reusability of processes at computational level*. The current practice in simulation-based design frameworks is to capture information about products and processes in a lumped fashion (i.e., the processes are captured in terms of the product information and the tools used), which restricts the utilization of processes for designing different products. There is very tight coupling between product and process information.

The requirement for the modeling approach is *modular use of processes for different products*. There is a similar requirement for modular information use - *modular use of processes for different design problems*. This requires the information model to facilitate utilization of the same design processes for different kinds of design problems.

Further, formal means for capturing design process networks in a computer interpretable manner are lacking. The reusability of design processes is currently limited to a human reading the process description. For designing design processes, the process models should be represented on a computer so that it is possible to *reuse sub-processes in new design scenarios* thereby facilitating the synthesis of processes. Our vision is to take this further and develop analyzable and executable design equation for specific design scenarios such as multi-disciplinary analysis and design.

7.1.2 Approach Proposed in this Dissertation – 3-P Information Model

In order to address design information modeling challenges stated in the previous section, we propose a strategy in this dissertation that consists of three main elements – *a)* modular systems view of design processes (discussed in Section 7.2.2), *b)* decision-based design (discussed in Section 7.2.1), *c)* separation of declarative and procedural information (discussed in Section 7.2.3). The strategy proposed in this dissertation is called as the *3-P information modeling strategy* (discussed in Section 7.3), which is embodied as an object-oriented information model that captures design related information (discussed in Chapter 8). The three key components of the 3-P model include information models for *a)* design ***problem***, *b)* design ***process***, and *c)* ***product*** information. The information model for design problem is discussed in Section 8.2, information model for product is discussed in Section 8.1, and for the process is discussed in Section 8.3.

The design *problem* definition consists of the design variables, responses, constraints, goals, preferences, etc. In order to solve the design problem, a design *process* is laid out, which consists of a network of transformations on product information. These transformations have *product* information at State A as input and product information at State B as output.

In the 3-P information model, the information is captured as entity objects and relationship objects. For example, in the models for design processes, the entities refer to information transformations and relationships refer to the information flows between these transformations, whereas while modeling product information, the entities refer to components and the relationships refer to interfaces between components. The design problem information model is based on the compromise DSP construct developed as a part of the Decision Support Problem (DSP) Technique (Muster and Mistree 1988). The processes are modeled as hierarchical systems in three levels – individual transformations, model interactions, and process compositions. Seven transformations and nine model interactions are identified specifically for simulation-based multiscale, multifunctional design. Each of these model interactions are associated with design processes and serve as standard reusable patterns for design processes. The information model is instantiated as Java objects and integrated into FIPER. The reusability and reconfigurability aspects of the information model are shown via modeling design problem, processes and product information for datacenter cooling system.

The 3-P information modeling strategy proposed in this chapter enables designers to capture design process information in a manner that allows quick process reconfiguration, thereby supporting design process exploration. The modular separation of information associated with problem, product, and processes enables exploring different design sub-processes for solving a given design problem. A prototype implementation of 3-P information model is carried out in ModelCenter, which is a distributed computing environment.

7.2 Proposed Strategy for Modeling Design Information

The strategy proposed in this dissertation for modeling design information is a synthesis of four key components – *a)* decision-based view of design processes and a specific instantiation – Decision Support Problem (DSP) Technique, *b)* modular systems based approach for design processes, *c)* mechanism for separation of declarative and procedural information, and *d)* an information model (3-P) for capturing design related information. Decision-based design is a conceptual model for design activities, which is based on the notion that the principal role of an engineer, in the design of an artifact is to make decisions (Mistree, Muster et al. 1989; Mistree, Smith et al. 1990). Decision-based design is chosen as a basis in this dissertation because of its domain independence (decisions are common across different engineering domains), phase independence (during any phase of the design process, the structure of decisions remain the same), and can be used for modeling any process in the value chain (not just design chains). A specific instantiation of decision-based design – the DSP Technique is chosen in this research as a basis for modeling design information, the details of which are discussed in Section 7.2.1. In order to support design of design processes, we view processes themselves as systems that consist of sub-systems interacting with each other through well defined interfaces. The interfaces for process entities are essentially the information flows into and out of the processes. Modular systems-based approach for design processes is employed in order to support reusability and composability of processes. This aspect of the proposed strategy is discussed in Section 7.2.2.

An important requisite for designing an appropriate design process for a given design problem is the ability to use different design processes for the same problem without the need to reformulate the design problem. As discussed in Section 7.1, the current methods and software tools capture the design problem and process related information in an integrated fashion, which increases rework and limits reusability. Hence, in this research, we separate the declarative information from the procedural information. The details of this modularity of information capture are discussed in Section 7.2.3. The embodiment of these ideas in the form of an information model is summarized in Section 7.3 and described in detail in Sections 8.2 through 8.3.

7.2.1 Decision-based Design and a Specific Instantiation: DSP Technique

The design model presented in this dissertation is an extension of the constructs developed within the DSP Technique proposed by Mistree and co-authors(Kamal, Karandikar et al. 1987; Mistree, Muster et al. 1989; Mistree, Smith et al. 1990; Bras 1991), rooted in the work of Simon.(Simon 1996) The DSP Technique consists of three principal components: a design philosophy rooted in systems thinking, an approach to identifying and solving Decision Support Problems (DSPs), and software. ‘Systems thinking’ encourages designers to view products and processes as systems interacting with the environment. In the DSP Technique, support for human judgment in designing is offered through the formulation and solution of DSPs, which provide a means for modeling decisions encountered in design. The DSP Technique allows designers to model design processes at various levels of abstraction (Kamal, Karandikar et al. 1987). The level of required software support is different at different levels of abstraction.

DSP Technique is a specific implementation of the decision-based design philosophy. In the DSP Technique, designing is defined as a process of converting *information* that characterizes the needs and requirements for a product into *knowledge* about the product. The DSP Technique consists of two phases, namely, meta-design and design (Muster and Mistree 1988). In the meta-design phase, the DSP Technique is concerned with finding an initial decision-based representation of processes (i.e., the information and knowledge of the design process). In the design phase, the DSP Technique is concerned with formulating and solving the Decision Support Problems (DSPs) in order to obtain implementable solutions (i.e., information and knowledge about the product). These two phases are completed in following six steps: *a)* identification of problem, *b)* partitioning and planning, *c)* structuring of DSPs, *d)* mathematical formulation of DSPs, *e)* solution of DSPs, and *f)* post solution analysis. The first two steps constitute the meta-design phase whereas the subsequent four steps constitute the design phase. These steps are shown in the Table 7-3.

Table 7-3 – Two phases and six steps of DSP Technique (Bras 1992)

Phase 1: Meta-Design	**Phase II: Design**
Step 1: Identify Problem	*Step 3 and 4:* Structure Support Problems
	Step 5: Solve Support Problems
Step 2: Partition and Plan	*Step 6:* Post solution analysis

A palette of entities is proposed in the DSP Technique, to model design information. These entities are domain independent and can be used to model hierarchies of design processes. The DSPT palette (shown in Figure 2-2) consists of three types of entities – potential support problem entities, base entities, and transmission entities. Potential support problem entities include phases, events, tasks, decisions and systems. These entities have Support Problems associated with them. Base entities are the most elementary entities that are easily implementable on a computer. Transmission entities are used to model connections between other entities. These entities capture inputs and outputs of other entities.

In the DSP Technique, support for humans is provided through Support Problems (SPs), especially DSPs. Phases, events, tasks, decisions and systems have associated SPs. Each SP captures information related to that entity in a structured format. The SPs are described in terms of key words and descriptors. The key words and descriptors are domain independent. An example of domain independent description of compromise Decision Support Problem (cDSP) is shown in Table 7-4. Similar other SPs are modeled for Selection decisions, tasks, events, phases, and systems. SPs are instantiated by populating domain dependent information for each keyword. The keywords and descriptors formalize the information that is required to completely model each SP. Since these keywords and descriptors are domain independent, they represent a common structure (conceptual schema) for SPs from any domain. This is one of the most important characteristics of the DSP Technique that enables reuse of design information across domains.

SPs can be modeled at various levels of abstraction – in terms of keywords and descriptors at the highest level of abstraction and in terms of base entities at the lowest level of abstraction. Hence SPs serve as medium between the human designers and computer implementation of design processes. The motivation in DSP Technique is to model SPs using the base entities on a computer. If a support problem contains all the information in terms of computational base entities, it is referred to as a template that can be executed on a computer.

Table 7-4 - Keywords and descriptors for compromise DSP

Keywords	**Descriptors**
Given	Symbolic and mathematical base entities and support problems necessary for evaluating the goals, constraints, bounds and deviation variables
Find	System variables
Satisfy	Goals, constraints, and bounds
Minimize	A deviation function

DSP Technique for modeling design information: From the perspective of computer supported modeling of design processes, the DSP Technique provides a framework for modeling, representation, manipulation and reuse of design processes on a computer. The notion of SPs, and its various levels of abstraction provides a mechanism to provide support for human decision making in a computational fashion. DSP Technique is also the only method that offers computational model of design processes in terms of design decisions. A designer working with the DSP Technique has the freedom to use sub-models of a design process created and stored by others and to create models of the intended plan of action using the aforementioned entities (Bras and Mistree 1991). DSP Technique has an underlying mathematical equation that can be used to mathematically model the design processes as a network of transformations.

What is missing from DSP Technique Palette? The drawback of the DSP Technique palette is that in its current form, it lacks an information model for representing product information. The DSP Technique palette is useful for capturing processes in terms of transformations but cannot be used in the current form to capture states of product information and their evolution. Further, only decisions are formalized so far in DSP Technique. Other transformations on product information along the design process are not formalized. For example, it is not possible to computationally represent abstraction, concretization, etc. using the DSP Technique palette. Hence, it is difficult to study the effect of transformations on product information. The DSP Technique palette needs to be extended to include product information and a close integration of product and process information needs to be established. This can be accomplished by infusing generic product information transformations in the DSP Technique palette.

Although the DSP Technique palette supports reuse of design processes, the reuse is mainly limited to pictorial and descriptive reuse. Reuse at a computational level is limited. The DSP Technique palette needs to be extended to incorporate modular and reusable entities that represent evolution of product through a series of transformations. Since each design process is unique, we believe that instead of modeling each design process individually, it is important to identify the basic types of activities performed during design. We are not so much interested in modeling each and every activity in the design process (the way current process modeling tools work for making organizational decisions). However, we are interested in modeling the transformations of product information from one state to another starting from requirements leading to the final design.

Augmentation of DSP Technique Palette in this dissertation: In order to address these limitations of the DSP Technique palette, we propose an augmented model for information capture, which is based on the philosophy of decision-based design. In this dissertation, we have modified the definition of design processes to *a network of transformations that convert product information from one state to another.* This definition of design processes is adapted from the work of Mistree and co-authors (Kamal, Karandikar et al. 1987; Mistree, Muster et al. 1989) where they define designing as a process of converting information that characterizes the needs and requirements for a product into knowledge about the product. This modification is proposed to eliminate the confusion in design community related to the difference in meaning of words *information* and *knowledge*.

From the requirements to the final product, design processes are carried out through a number of phases. For example, the phases associated with Pahl and Beitz (Pahl and Beitz 1996) design process are - planning and clarification of task, conceptual design, embodiment design and detailed design. Each phase is associated with *stages* of product information and converts information from one *stage* to another. Within each phase, there is a network of transformations that operate on product information. These transformations can be carried out in a sequential (as shown in Figure 7-3) or parallel fashion (not shown). The transformations operate on product related information and convert this information from one *state* to another. The state of information refers to the *amount* and *form* of that information that is available for design decision-making. For example, *analysis* is a transformation that maps the product form to behavior, whereas, *synthesis* is a mapping from expected behavior to the product form. It is important to note that these transformations remain same during different phases of the product realization process, as shown in Figure 7-3.

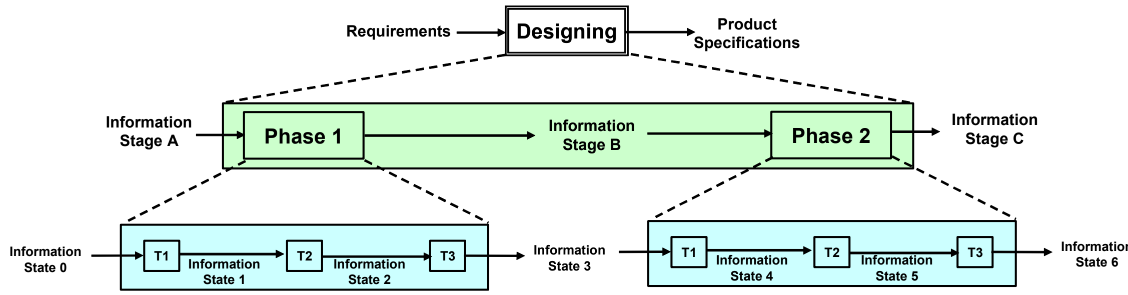


Figure 7-3 - Sequential design process as a series of transformations

In this chapter, we build the information model starting with this view of design. Each of the transformations transforms the product information from one state to another. As in the DSP Technique, each transformation is associated with a Support Problem, termed as Transformation Support Problem (TSP). Decision Support Problems (compromise and selection) are special types of transformation support problems. Other TSPs are identified and formulated in Section 8.2. Analogous to the DSP Technique, the design method consists of two phases – metadesign and design. Meta-design phase is concerned with formulation of TSPs, whereas the design phase is concerned with solution of the TSPs. The solution of TSPs takes place through a network of tasks – the design process. The tasks can either be simple tasks that can be executed directly (such as execution of a finite element code) or can be transformations (that require formulation as TSPs). The transformations are further associated with a design process, representing the hierarchical nature of design. It is an assertion that transformations in a design process are same at different levels in the hierarchy.

The core element of the proposed model of designing, as described above, is shown in Figure 7-4. As shown in the figure, there is a separation of information related to formulation of TSPs (declarative information) and the information related to their solution (procedural information). The details of this separation of declarative and procedural information are discussed in Section 7.2.3. In order to model the design information, the three key elements – Transformation Support Problems (TSPs), product states, and design processes should be modeled. An information model for these key elements is presented in Section 7.3. The core element of designing shown in Figure 7-4 is an important building block that can be used over and over again for modeling design processes at any level of abstraction and for any domain. This embodies the concept of modular systems view of design processes. This aspect of modular systems approach is discussed next in Section 7.2.2.

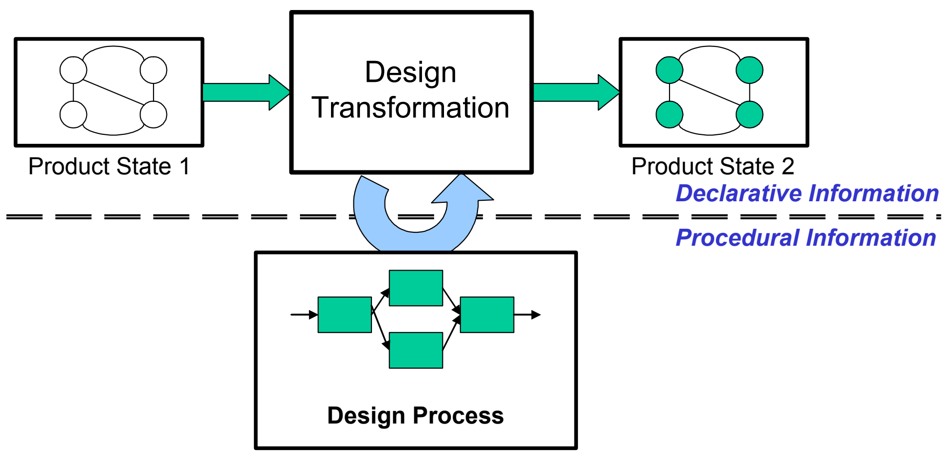


Figure 7-4 - Proposed model for designing – transformation of information from one state to another

7.2.2 Modular Systems Approach for Design Processes

One of the main challenges in modeling any design effort, regardless of scale or scope, is standardizing the manner in which information and associated dependencies are represented. The need for reusability of information translates this requirement into representing information in a domain neutral form that supports designers in providing and structuring required information content in a computationally archivable and reusable manner. This calls for a domain independent means of capturing design information. In order to facilitate designer interactions required for effective collaboration from a decision-based perspective, expression of design decision related information in a standardized format is also required. It is for this reason that we advocate a *modular template-based approach* to modeling design information. A *template* is commonly defined as (1) a pattern, used as a guide in making something accurately, (2) a document or file having a preset format, used as a starting point for a particular application so that the format does not have to be recreated each time it is used.[4] Clearly, the word *template* is appropriate in our context because it implies reusability, achievability, and support/guidance.

In order to effectively support engineering design processes, this notion translates to the development of reusable computational templates for design. These computational templates should serve as building blocks – completely modular components that are standardized with respect to structure and interface architecture. Such building blocks must also facilitate analysis, and execution. Currently, there is a lack of formal, executable, computational models for representing and reusing existing knowledge about design processes. The only knowledge that is readily available is confined either to designers' expertise or to descriptive/pictorial forms of documentation. This is a result of the predominantly narrative or symbolic nature of current models.

Our design process modeling strategy is based on the assumption that processes themselves are hierarchical systems that can be progressively broken down into sub-processes that in turn can be represented in terms of basic design process building blocks, namely the *information transformations*, discussed in the previous section. Specifically,

[4] Compiled from *www.dictionary.com*

we focus on developing modular, reusable models of information transformations with clearly defined inputs and outputs that facilitate hierarchical modeling of design processes. Due to their consistent structure, design processes modeled in this fashion provide the ability to easily archive and reuse design process knowledge at all levels of the model hierarchy.

The fundamental concept of constructing process templates from networks of design process building blocks is illustrated in Figure 7-5. The design process in this figure involves three information transformations, namely, T1, T2, and T3. Each of these templates is at a different level of completion. T1 is a complete template, implying that all the information required for its execution is available. T3 on the other hand has yet to be instantiated relevant to the problem at hand and consequently, does not differ from generic information

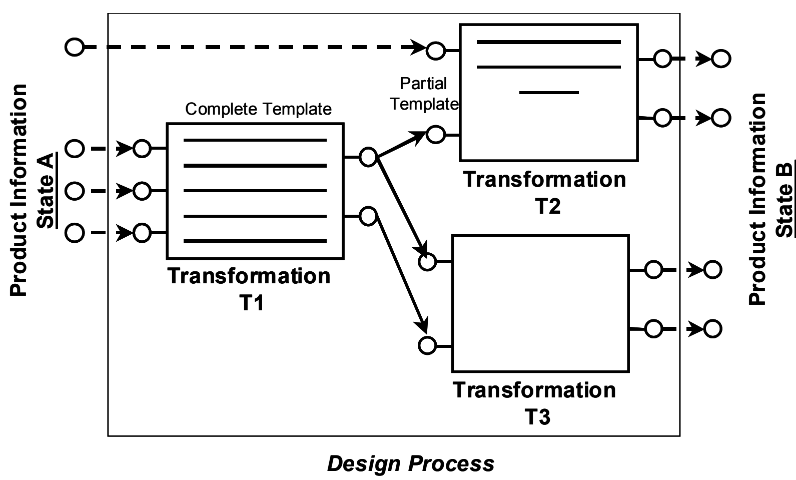


Figure 7-5 - Modeling design process using process templates

In order to facilitate reuse of design process models, the building blocks of design processes must be generic (domain independence) and modular. We aim to facilitate design process reuse with respect to (1) hierarchical composition and (2) cross-domain application, respectively. The underlying relationship between these two dimensions is illustrated in Figure 7-6. Domain independence of decision templates is derived from the underlying DSP Technique constructs, as described in the previous section. The DSP Technique palette contains various entities such as phases, events, decisions, tasks, and systems (Mistree, Smith et al. 1990) for modeling design processes. Since there is a support problem associated with each DSP Technique palette entity, the use and reuse of design process models and design sub-process models, created and stored by others, is thus facilitated. Due to the domain independence of the underlying constructs and the integrated systems perspective, the DSP Technique offers a solid foundation for developing computational models of reusable design processes, as envisioned in this dissertation. Their hierarchical composability emanates from the novel application of modularity principles to design process building blocks.

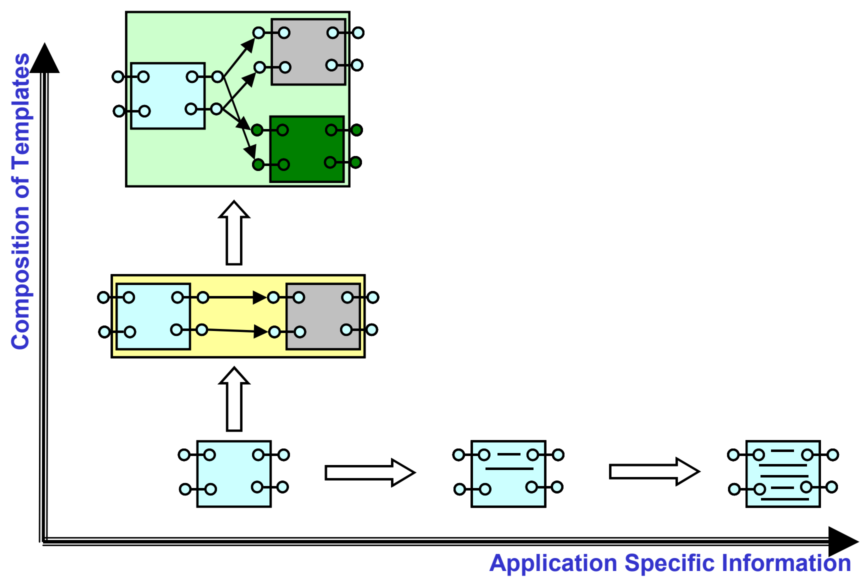


Figure 7-6 - Reusability of design processes with regard to hierarchical composition and cross-domain application

It is the nature of the information content, captured within these templates, that serves as the only differentiator among instantiated constructs; the underlying structure remains the same regardless of context or application. Having outlined the modular systems-based perspective espoused in this research, we proceed to discuss the idea of separation of *declarative* and *procedural* information in Section 7.2.3. The templates are defined based on separation of the declarative and procedural aspects of design information, resulting in generic information transformation constructs that are instantiated as software templates.

7.2.3 Separating Declarative and Procedural Information

The current state of information models and design support tools force designers to think in terms of the *procedure* for solving a particular problem rather than conceptualizing and *declaring* the problem itself that they want to solve. We believe that the separation of this declarative and procedural information is extremely important for development of effective design support systems. Referring back to Figure 7-4, the extension of DSP Technique used in this dissertation categorizes design information into two types – declarative and procedural information. The information associated with design transformation and the product states is declarative information because it refers to what is done by the designer through that transformation. The manner in which this information transformation is carried out is procedural information because it refers to how that transformation is carried out through a network of tasks. Declarative information captures all the pieces of information / knowledge and relationships between them that represent the transformation to be carried out. After the designers have declared their design problem, it can be executed using many different processes. Configuration of the right process for that problem is the challenge in designing design processes.

The idea of separation of declarative and procedural information is analogous to understanding the behavior of system that is represented by a set of linear equations. The first step for understanding the system behavior is formulating (*declaring*) all the equations that correspond to the information/knowledge available to designers. After the equations are formulated, the next step is to select a *process* to be used for solving those equations simultaneously. Various algorithms (that correspond to the *processes* for solving the equations) such as Cramer's rule, Gaussian elimination, LU decomposition, Jacobi method, etc. are available for solving set of linear equations. Appropriate selection of algorithms (process) depends on the characteristics of linear equations such as diagonal dominance, sparcity of the matrix, etc. The selection of right process is analogous to designing the design process for executing a design transformation.

One of the advantages of separating declarative and procedural information is that this scheme forces designers to focus on design problem formulation first rather than its solution. This is important because without appropriately formulating the design problem, the designers are likely to ignore important considerations for designing. The second advantage is that it enhances reusability of the design processes for solving different kinds of design problems. The third advantage is that it supports design of design processes in a systematic manner.

7.3 3-P Information Modeling Strategy for Problem, Product and Process Information

The 3-Ps refer to the key elements of design information – Problem, Product, and Process. The 3-P information model is developed to support development of tools that support designers in designing both products and associated design processes. This is allowed through the modular plug-and-play of different processes to different problem formulations. The 3-P information model is an instantiation of the concepts presented in Sections 7.2.1 through 7.2.3. Extension of the DSP Technique is used to define the design transformations that map the product information from one state to another. The information about transformations and product information is declarative information, whereas the information about processes is procedural information. This achieves separation of declarative and procedural information. The 3-Ps – *a)* problems associated with transformations, *b)* product information, and *c)* design processes are captured as generic templates. These generic templates can be instantiated by populating information specific to the problem, product, or the process (see Figure 7-7). Due to the inherent independence with which each of these three elements is described, instantiated templates provide the required modularity in the information architecture. Modularity in this information representation allows configuring different problems with different processes and applying these for variety of product design scenarios.

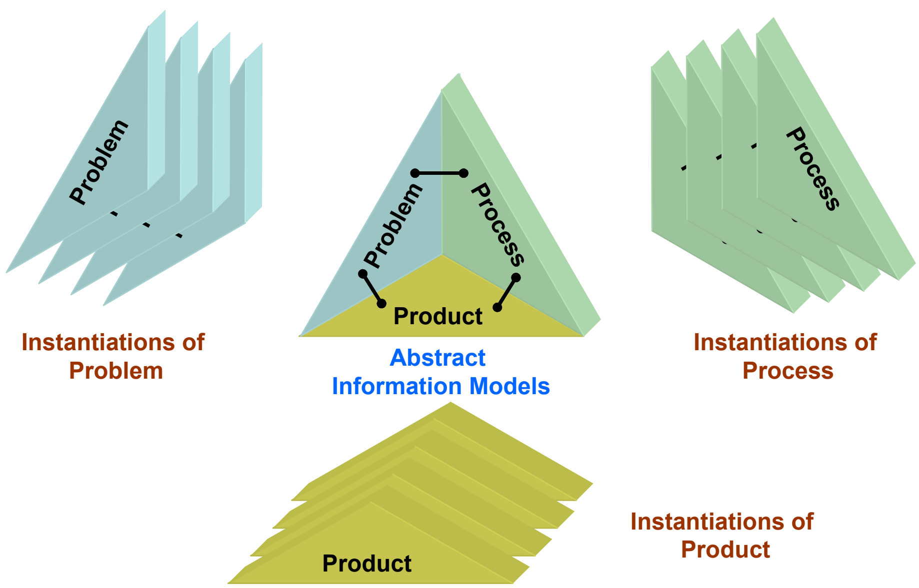


Figure 7-7 – Different instantiations of a common abstract information models

The notion of combining different products, problems, and processes together is illustrated in Figure 7-8. In this figure, five different problem formulations, including Archimedean formulation, utility based formulation, robust design formulation etc., are shown for the decision support problems. Three different product instantiations – pressure vessel, datacenter, and multiscale materials are shown. Four different types of processes are listed for executing the decision support problem. The 3-P information allows different combinations (as shown by connecting lines in the figure) of instantiations of 3-Ps. This implies that instantiated templates for Archimedean formulation of decision support problem can be instantiated for pressure vessel design using the process corresponding to Robust Concept Exploration Method (RCEM). Similarly, the problem formulation using Design Capability Indices (DCIs) can be instantiated for pressure vessel design using the process corresponding to interaction patterns.

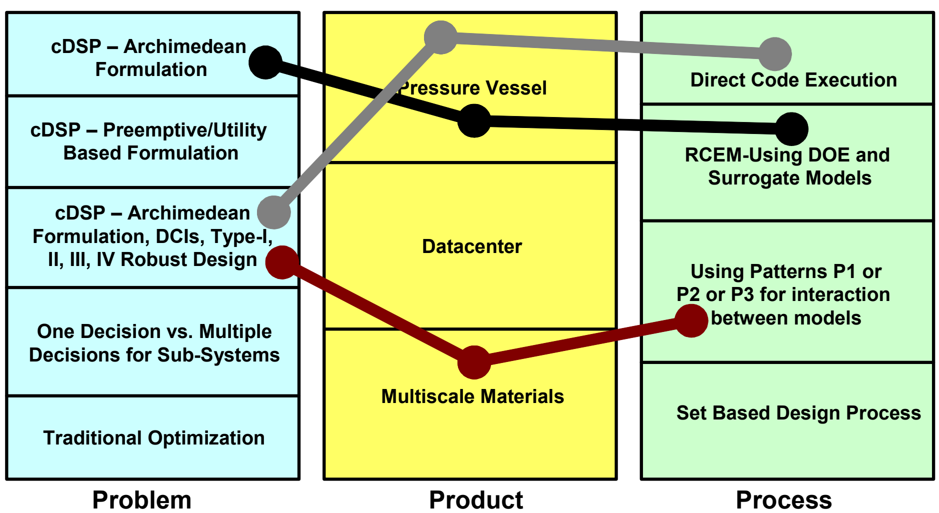


Figure 7-8 - Different combinations of problems, products, and processes

How is the 3-P information model embodied? The 3-P information model is implemented using object oriented constructs because they support hierarchical relationship between entities and the support reuse. The information model is instantiated as abstract Java classes. In order to support different levels of abstraction of information, these generic templates are described at different hierarchical levels. For example, the most abstract compromise decision support problem is defined with basic components –

design space, response space, preferences, etc. The preferences can be further defined as simple Archimedean preferences, preemptive preferences or rather more complex utility based preferences. These preference structures dictate different levels of hierarchical templates for the compromise decision support problem.

The abstract Java classes are referred to as the generic templates that can be instantiated for specific scenarios and combined together to generate an executable description of the design process that is particularized to design a specific product by solving a specific problem. The instantiation of generic templates into specific templates is carried out by extending the abstract Java classes. Specific schemas for the 3-Ps are presented in Sections 8.2.2, 8.1.2, and 8.3. The details of instantiation of generic templates are also discussed in these sections using simple examples from pressure vessel and spring design. In addition to the independent use of this information model, the augmentation of currently available commercial tools such as iSIGHT (2004), FIPER (Engenious Inc. 2004) and Model Center (Phoenix Integration Inc. 2004) is also discussed.

Advantages of 3-P information modeling Strategy

The *3-P* modeling approach, proposed in this dissertation, enables designers to capture design process information in a manner that allows quick process reconfiguration, thereby supporting design process exploration. The modular separation of information associated with problem, product, and processes enables exploring different design sub-processes for solving a given design problem. The key advantages of the *3-P* approach arise from the three basic ideas used for its development (extension of DSP Technique, modular template based approach, and separation of declarative and procedural information). These advantages include the following:

1. Information can be modeled at different levels of abstraction due to the utilization of object oriented constructs
2. Information related to Problems, Products and Processes is separated and captured via modular templates
3. Different combinations of Problem, Product, and Process declarations can be combined together to generate specific computationally executable processes
4. Process knowledge can be captured and reused across problems and products
5. The information model allows composability of instantiated sub-processes into higher level processes.

In addition to the independent use of the proposed approach, it can serve as an augmentation of currently available commercial tools such as iSIGHT (2004), FIPER (Engenious Inc. 2004) and Model Center (Phoenix Integration Inc. 2004).

Steps in Utilization of 3-P Information Model

Five steps in utilization of 3-P information models are shown in Figure 7-9. In each of the steps, an element of information is created, transferred, or updated. In the first step, the designers select blank (un-instantiated) templates for product, process and the problem. The dotted line between the problem and the process templates represents that the abstract Java classes for process are defined to exchange information with the abstract Java classes that define the problem. In the second step, designers instantiate the product information – i.e., provide ranges for design variables, and values for parameters, etc. This is represented by one filled oval in the product information. The information available at this step is labeled State A. The designers are then able to instantiate the problem definition in Step 3. Instantiation of the problem template requires designers to utilize information about attributes and relationships defined in the product information. This information flow is shown as solid line from instantiated product template in Step 3. In addition to the product specific information, there is also additional problem related information such as constraints, targets for goals, and designers' preferences for goals. After the problem is defined, this information can automatically be transferred to the un-instantiated process template to generate an executable process description. The generation of executable process description is shown in Step 4. The execution of this instantiated process description results in additional information about the product, which is used to update the product information from State A to State B as shown in Step 6.

The steps shown in Figure 7-9 are for execution of a single design transformation using the 3-P information modeling strategy. A general design process consists of a network design transformations (refer Figure 7-3), that transform the product information from one state to another. For example, in a scenario with two sequential transformations, there are three states of product information (see Figure 7-10). Each of the transformations is associated with support problem templates that can further be associated with different process templates. The process templates can be instantiated to generate executable processes.

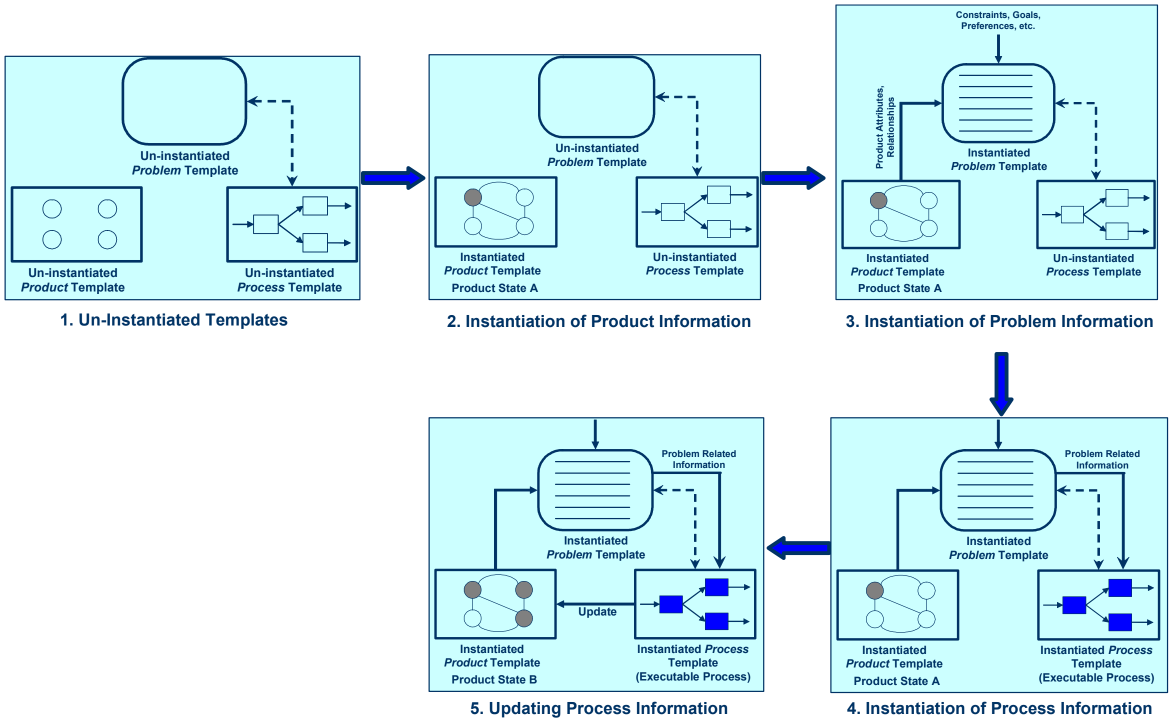


Figure 7-9 - Five steps in utilization of 3-P information model for a single design transformation

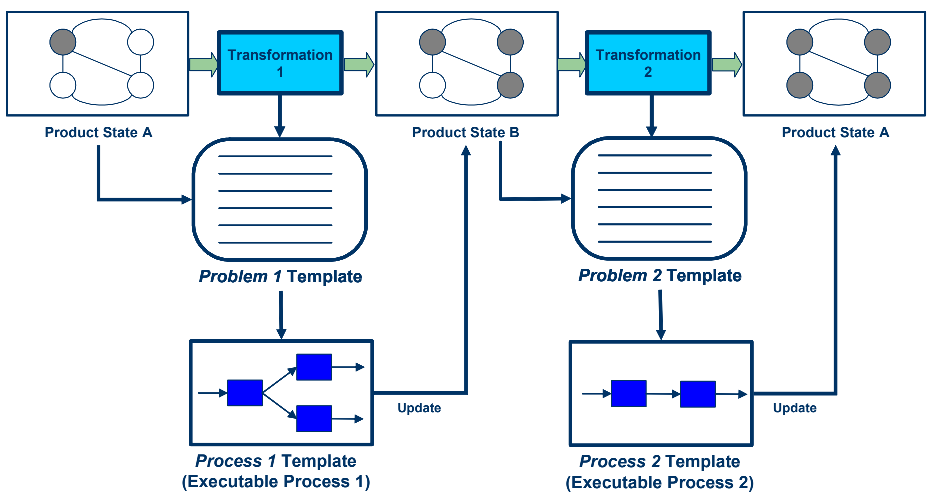


Figure 7-10 - Utilization of 3-P information model for multiple transformations

7.4 Role of Chapter 7 in the Dissertation

In this chapter, we present a *strategy* for modeling information in a manner that supports designing design processes. The strategy is based on the hypothesis that separation of product, process, and problem related information in a modular fashion supports utilizing same design processes for different design problems and different products. The details of implementation of this information modeling strategy are presented in the following chapter (Chapter 8).

Chapter 8 Implementation of the Proposed Design Information Modeling Approach

The focus in this chapter is on providing the preliminary implementation details and validation of the information modeling strategy presented in Chapter 7. The implementation of the 3-P information modeling strategy involves developing separate information models for products, problems, and processes. The information model for products is discussed in Section 8.1. Information model for problems is discussed in 8.2 and the information model for processes is discussed in Section 8.3. Each of these sections is structured such that the general concepts to be included in the information model are discussed, following a schema consisting of the concepts and relationships between them. The general strategy of 3-P approach is validated via implementation in ModelCenter, which is a commercially available simulation-based design framework. The implementation is tested for design of two different products (design of a pressure vessel and design of a spring) using same design process and problem structure that is stored as templates. It is important to note that the implementation in ModelCenter is not a complete implementation of the information models presented in Sections 8.1 through 8.3, but it provides simple validation of the general concepts underlying the 3-P approach. The hypothesis and validation examples discussed in this chapter are highlighted in Figure 8-1. The validation is performed using pressure vessel and spring design examples.

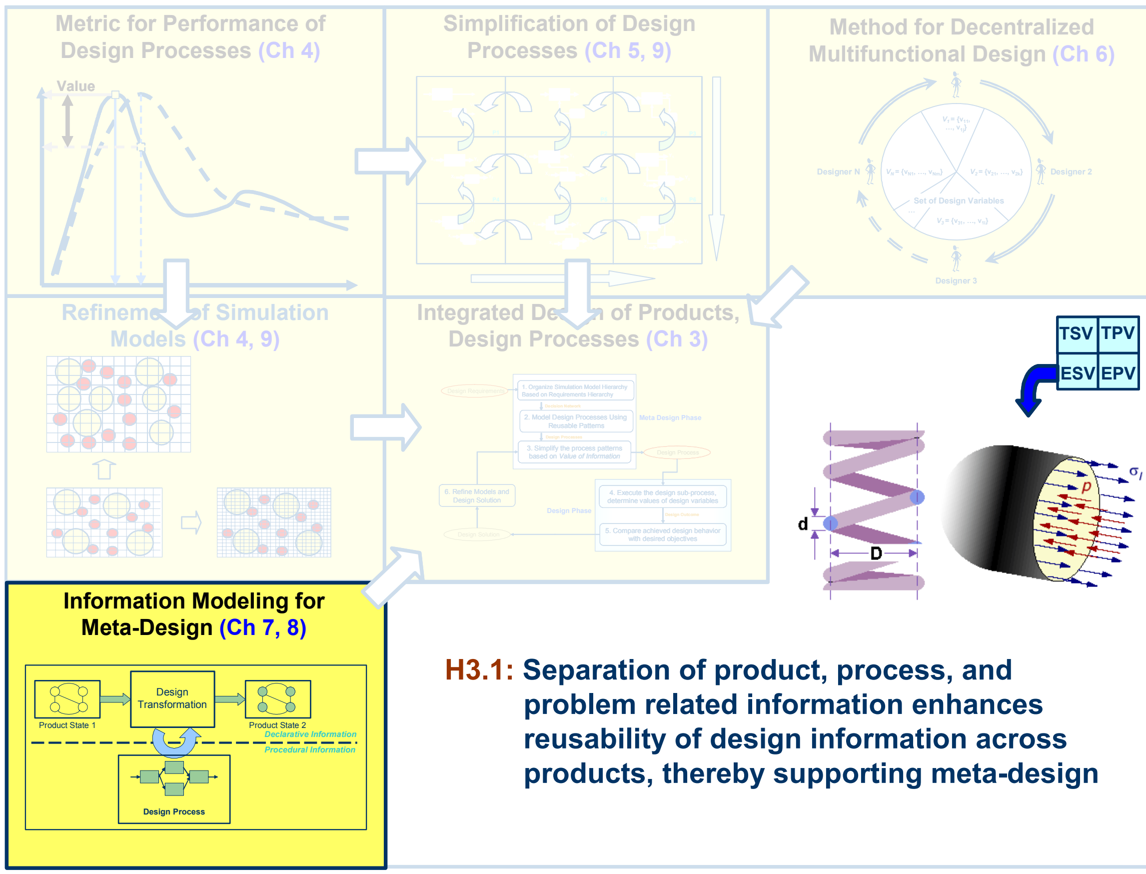


Figure 8-1 – Hypothesis addressed in Chapter 8

8.1 MODELING PRODUCT INFORMATION – OPERANDS IN DESIGN EQUATION

One of the three components of the 3-P information model is the product information. The key requirements for the product model of 3-P information model as pointed out in Section 7.1.1 include – *a)* mathematical form associated with the information model, *b)* representation of all alternatives under consideration during a given point in the design process, *c)* ability to capture relationships between sub-components, *d)* ability to capture evolution of product information, *e)* ability to capture information at various levels of abstraction, *f)* ability to capture different types of design process scenarios, *g)* ability to support human decision making, and *h)* ability to support simulation-based multifunctional design. The product model presented in this section addresses these requirements.

Any mathematical equation is defined in terms of *operators* and *operands*. In order to model the design equation in mathematical terms, we need to define both the operands and the operators that occur in design. As discussed previously, the design transformations (that serve as operators) in design equation act on the product information. Hence, the operands are necessarily defined in terms of the product information. In this section, we present our information model for representing product information that is based on set-theoretic principles, and acts as the operands in the design equation. The product model presented in this section forms a basis for modeling transformations in design processes.

8.1.1 Proposed Mathematical Product Model – Operands in the Design Equation

Before presenting the information model, we provide definitions of the keywords used.

Definitions

1. An *entity* is an abstraction of a physical object, concept or phenomenon. An entity can be a collection of other entities. For example, the entities associated with datacenter are room, computer, processor, rack, etc. that can be represented with their dimensions and locations. A complete datacenter is also an entity that can be represented with an array of racks with their relative positions.
2. *Attributes* are entities that describe a particular aspect of another entity.

3. A *parameter* is a special type of attribute that takes as value a Real number. For example, length, width, height, air velocity, air density, heat transfer rate, etc. take Real values and hence, are parameters.
4. A *relationship* is an association between entities or attributes.
5. The *form* of an entity is defined as a collection of attributes that can be controlled directly by the designer. For example, overall dimensions are the form parameters of the LCA entity. Dimensions of individual voids are form parameters of void entity.
6. The *behavior* of an entity is defined as a collection of attributes that describe the product's functionality. The behavior of a product can be derived from the product's form and its interactions with the environment. The behavior can be modified by changing the form attributes. For example, stiffness and overall heat transfer are parameters related to behavior or LCAs.
7. The *state of product information* represents the combination of form space and the behavior space at a given point in time.

Having defined the key terms used in the product model described in this section, we now move on to the product model.

Product Model

The product model adopted in this section is based on object oriented information modeling concepts. In our work, product model Π is represented as a set of entities (ε_i), relationships between these entities $(\rho_{i,j})$, and relationships with external entities (ρ_k).

$$\Pi = \{\varepsilon_i, \rho_{i,j}, \rho_k\}, \qquad i, j = 1..n, \;\; k = 1..m$$

This model is illustrated in Figure 8-2 where three entities $\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ representing the product Π are shown. These entities are related to each other through relationships $\{\rho_{1,2}, \rho_{2,3}, \rho_{1,3}\}$. The relationships of this product Π are represented as $\{\rho_1, \rho_2\}$.

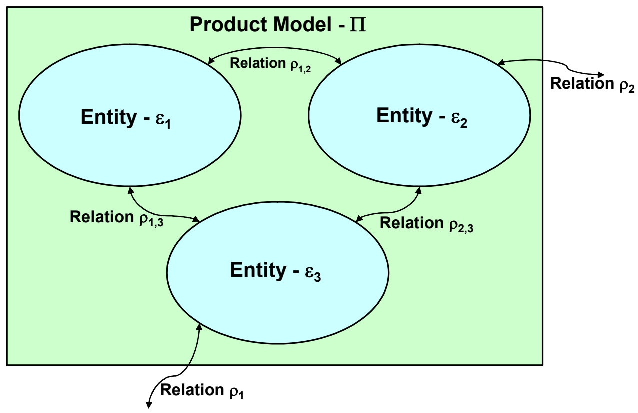


Figure 8-2 - Product model as a set of entities and relationships

Hierarchy in Entities

Each entity (ε_i) can further be represented in terms of other sub-entities (ε_{ij}) that describe different aspects (or attributes) of the entity being represented. This enables representation of information in a hierarchical form. This hierarchical form of entities is shown in Figure 8-3. The entity (ε_1) is represented in terms of sub-entities $\{\varepsilon_{1,1}, \varepsilon_{1,2}, \varepsilon_{1,3}\}$. Similarly, entities (ε_2) and (ε_3) are represented by corresponding sub-entities. It is important to note that the sub-entities may or may not be subcomponents of the product. The sub-entities may represent information about different aspects of the product.

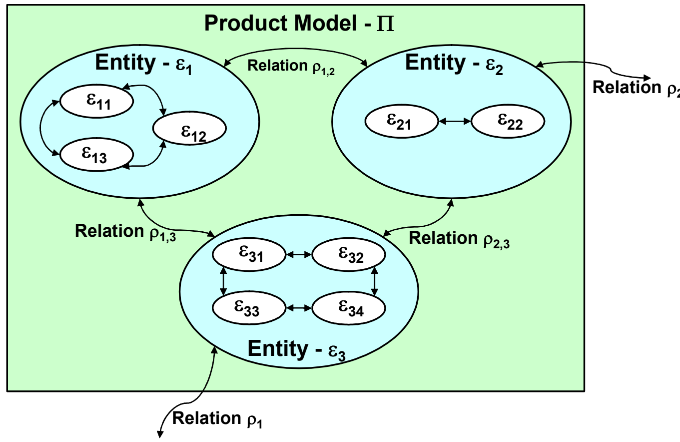


Figure 8-3 - Hierarchy of entities in product model

Entity Set

The entities (ε_i) can assume a set of values (ε_i^k) that represent different embodiments of the entities. A set of values associated with an entity is termed as an *entity set.*

$$\varepsilon_\imath = \{\varepsilon_\imath^1, \varepsilon_\imath^2, .., \varepsilon_\imath^k, ..\varepsilon_\imath^m\}, \qquad k = 1..t$$

These entities and associated sets of values are represented in the form of a space where each entity represents an axis (dimension) in the space and the set of values represent the values on that axis. The set of possible values can be discrete or continuous. The product space is illustrated in Figure 8-4 for three entities and associated alternative values. A point in the product space represents an instantiation of a product. Different points in the space refer to different products.

The simplest entities are the ones that cannot be broken down further in terms of sub-entities. These entities are called *parameters*. The parameters take values on the Real line. Hence, the Real line is the entity set for parameters. The product space is mathematically represented as a cross product of entity sets. In other words, a representation of product form consists of an *n-tuple of entity sets*.

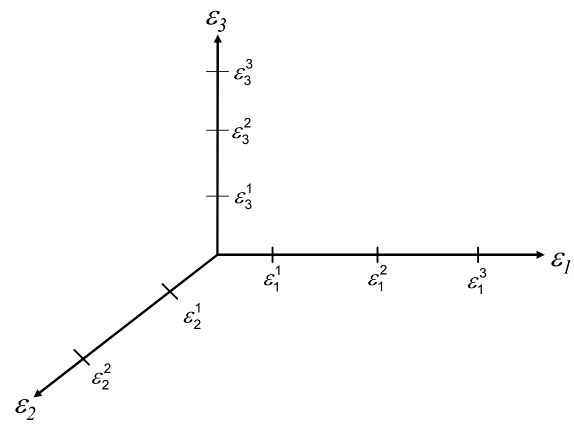


Figure 8-4 - The product space represented by entities and their alternative embodiments

Relationships in product model

In addition to the entities and associated attributes, relationships $(\rho_{i,j})$ between entities and constraints form an important component of the product model. Constraints are special types of relationships in the product model. These relationships and constraints represent surfaces or sub-spaces in the product space. The relationships in the context of product model presented in this dissertation are mathematical in nature. Relationships are surfaces if the mathematical form is represented as an equality relation, whereas the relationships are sub-spaces if they are represented by inequality relationships. For example, in the datacenter product model, lower bounds on the size of the overall dimensions (Length > $\text{Length}_{\text{lower bound}}$, Width > $\text{Width}_{\text{lower bound}}$) are sub-spaces in the form space. Similarly, the maximum allowable temperature on the surface of processors and the bound on average temperature divide the design space into feasible and infeasible design spaces. The relationship between air flow rate, velocity and cross-sectional area is equality and hence, represents a surface in the product space.

Behavioral models relate entities in the form space with entities in the behavior space. Constraints separate the overall product space into feasible space and infeasible space. Feasible product space refers to the collection of points where constraints are satisfied and the infeasible space refers to the points where constraints are not satisfied.

Form and Behavior Spaces

The product space is divided into two sub-spaces – *form space* and *behavior space*. The form (entities that the designers can control) of the product represents a multi-dimensional space, called the *form space*, where each dimension represents an entity set. The form space of design is defined by the various dimensions, number of computers in a rack, flow rate of cold air, and the temperature of cold air into the room. The datacenter behavior relevant to the design scenario described in previous chapters consists of two parameters: average temperature on the computers, and maximum temperature on the processors. These two parameters represent the *behavior space* of the datacenters. Points in form space are *related* to corresponding points in the behavior space through physical laws. For example, the steady state temperature at the top of each processor can be directly evaluated from the form parameters. In a design process, the design requirements determine the desired product behavior and the designers' objective in the design process is to search for a point in the form space that corresponds to a required point in the behavior space.

8.1.2 Schema for the Proposed Product Model

In order to model the product information in a computer-interpretable manner, an object oriented schema is presented in Figure 8-5. The key part of this information model is an entity. A product is composed of many such entities. Each entity, in turn, is composed of various sub-entities and is defined by a number of attributes. Each attribute can be either of type form or behavior. Form attributes combine together to represent the form of an entity, whereas behavioral entities taken together represent the behavior of that entity.

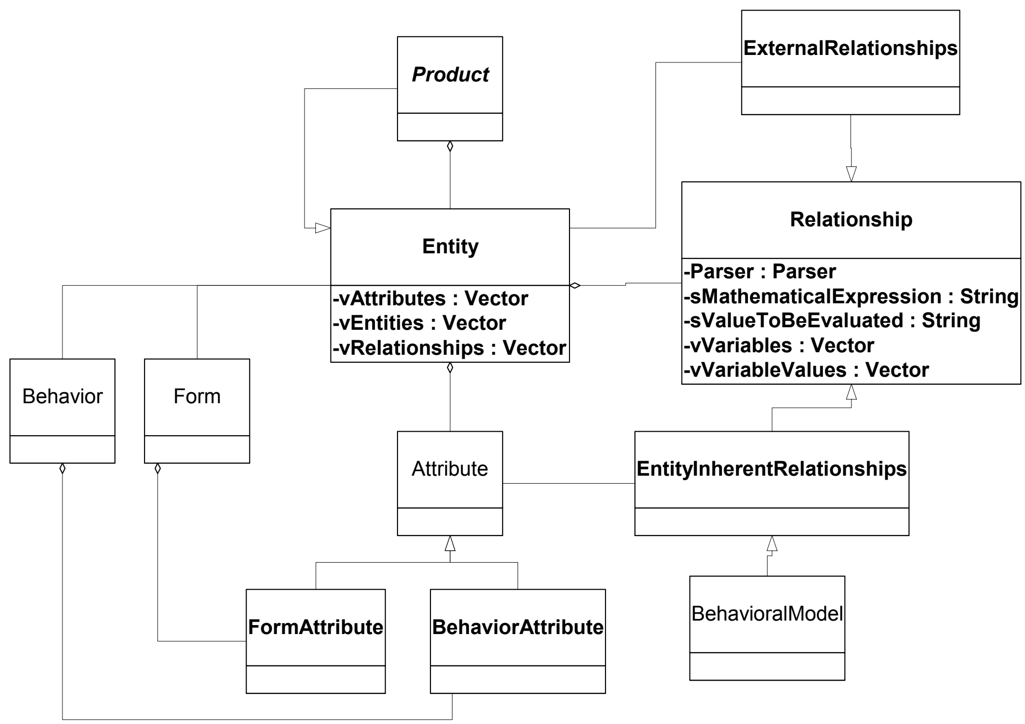


Figure 8-5 - Schema for capturing product information

Relations can be of two types – entity inherent relationships and external relationships. Entity inherent relationships are the relationships between the sub-entities of the entity under consideration. External relationships are the relationships of an entity with other entities. The behavioral model is a special type of entity inherent relationship.

It is reiterated and emphasized here that entities are abstractions of reality. Entities do not necessarily represent components of a product. These can be abstract concepts such as elements in Finite Element Model, boundary conditions, etc. Hence, the product hierarchy does not necessarily correspond to the part/subpart (assembly) hierarchy. The hierarchy represents a design perspective that the designer is interested in. This is different from most of the commonly used product models. The abstract nature of entities allows us to view attributes as special types of entities. Since the product information is defined by the designers based on their perspectives of design, different designers may model the same product in an entirely different manner.

8.1.3 Examples of Product Specific Information

The first example is from the design of datacenter cooling system. This example is discussed in details from the standpoint of integrated design of products and design processes in Section 5.3.2. Datacenters are huge computing facilities that house arrays of computers stacked in vertical racks. These racks are arranged inside the facility in a manner that allows easy access to all these computers and also facilitates effective removal of heat from these computers. During the design of a datacenter cooling system, the components of the system considered include the location and arrangement of racks in the room, arrangement of computers in each rack, and the temperature and velocity of air coming out of the air cooling system. The product model shown in Figure 8-6 has a hierarchy of entities, with the top most entity for datacenter. The datacenter entity is composed of entities for cabinets (racks) and airflow. Each cabinet consists of multiple computers, each of which is further composed of multiple processors. The airflow consists of two entities – air inflow and air outflow. The attributes associated with each entity are shown at the bottom of each entity. For example, the entity computer is associated with attributes including number of processors, height, average temperature, fan curve, etc. The relationships are shown using bi-directional dotted arrows and are labeled from 1 through 4. The relationships include the distance between cabinets, the temperature profile as a function of air inlet conditions, etc. Although it is not marked in the figure, some of the entities such as various dimensions, number of processors, computers etc. are form attributes and the others such as maximum and average temperatures, velocity vectors, etc. are behavior attributes. Although this is not a complete information model for the complete datacenter, it serves as a simple example for concepts discussed in Section 8.1.1.

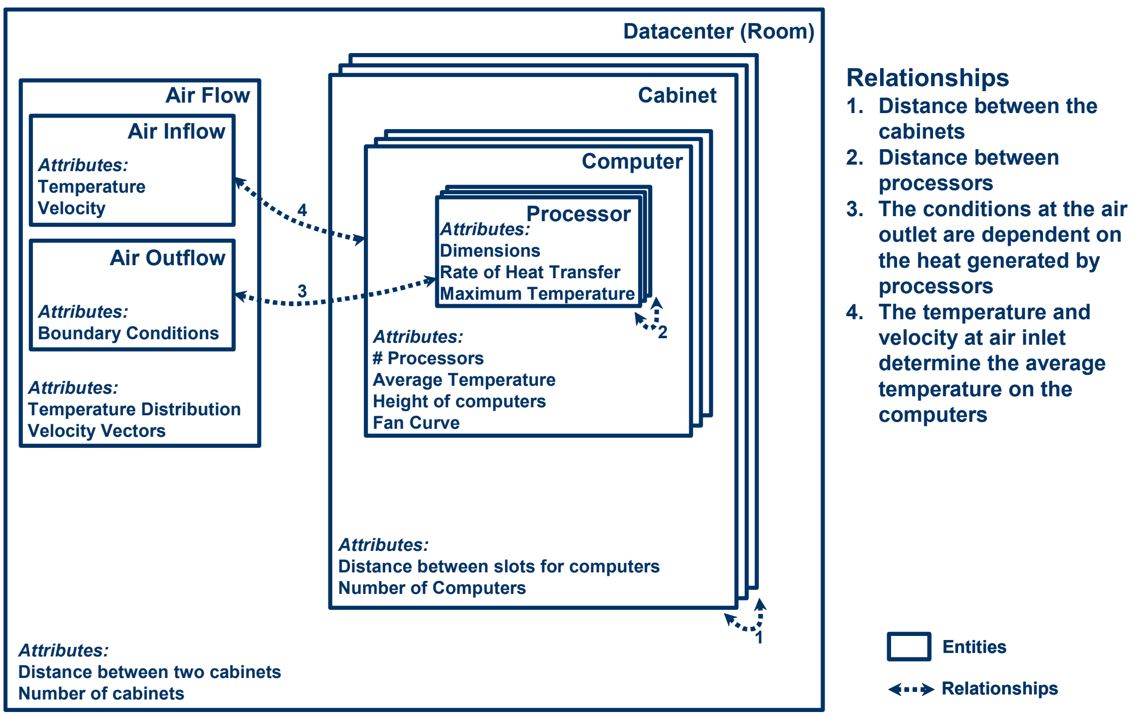


Figure 8-6 - Product information for datacenter example

The second example is from the integrated design of products (such as a projectile for defense application) and materials. The entities in materials design information include the product, the material mixture microstructure, and the mixture constituents. The interface between materials is important in this design example and hence is modeled as a separate entity. The attributes for mixture include the volume fraction of different constituents, particle spatial distribution, etc. These attributes are related to the form of the material and can be varied by the designer to achieve desired behavior. The behavior related attributes include equation of state, stresses, temperature and pressure distribution. It is important to note that some of the relationships are simple mathematical relationships between parameters, whereas other relationships require execution of complex analysis codes such as finite element analysis. The details of the product information are not discussed here because it is not the focus here.

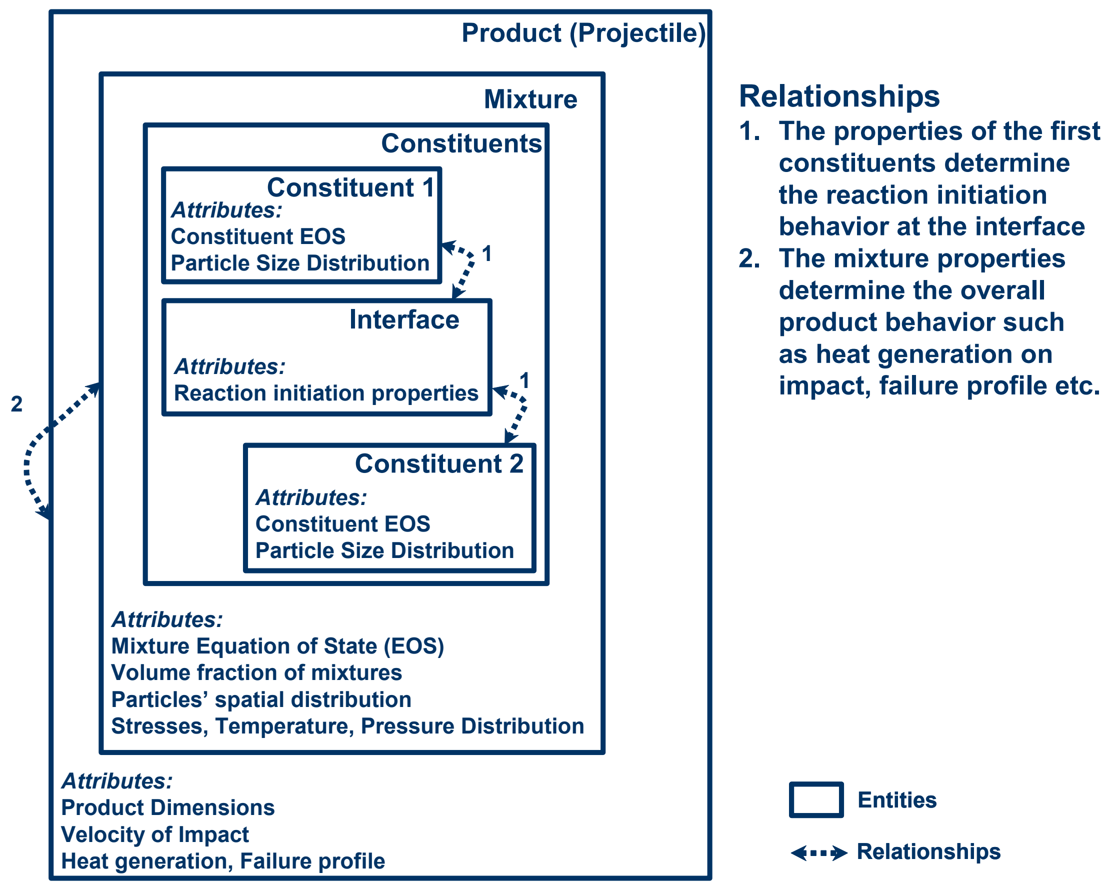


Figure 8-7 - Product information for materials design example

8.2 Modeling Design Problems – Transformations in Design

The second important component of the 3-P information model is information related to Design Problems. In this section, our focus is on extending the DSP Technique palette by modeling the fundamental information transformations encountered in engineering design. These transformations include *decisions, abstraction, composition, decomposition, interfacing, mapping, and synthesis*. Each of these transformations is associated with a support problem. In this section, we present the support problems associated with these transformations and structured according to the overarching systems model envisioned in the DSP Technique (discussed in Section 7.2.1). These support problems serve as generic templates for capturing problem related information in a declarative manner. Our approach is *decision-centric,* i.e., design tasks generate information that is ultimately useful for design decision making. Modeling a design process using such a decision centric approach involves developing networks of transformations. Modeling the design problems in a declarative fashion is an important part of mathematically modeling the design equation. We describe these transformations and associated support problems in detail in Section 8.2.1.

8.2.1 Proposed Transformations in Design

The key transformations associated with a design process include: Mapping, Decomposition, Composition, Abstraction, Refinement, Evaluation, and Decisions. In this section, we discuss the details of these transformations.

1. *Mapping: Mapping is a transformation that involves establishing relationships between different entities of a product model.* For example, in the Pahl and Beitz design process, the mapping of requirements to appropriate function structure, and the mapping of functions to working principles are two types of mappings in design processes.

In a simulation-based design process, the mappings are created between different sets of parameters that represent the entities. The most common type of mapping encountered in design is analysis. *Analysis* is a mapping of parameters from the form space to parameters in the behavior space. Analysis is carried out using physics-based behavior models. Another type of mapping that is very important from the decision-based design standpoint is mapping from the behavior space to the preference space. The preference space represents designers' preferences for required behavior and can be defined using various mathematical techniques such as deviation from the goals, utilities for behaviors, etc. This mapping is referred to as *preference evaluation*. *Synthesis* is a mapping from the parameters in preference space to parameters in the form space. These three mappings in design processes are shown in Figure 8-8.

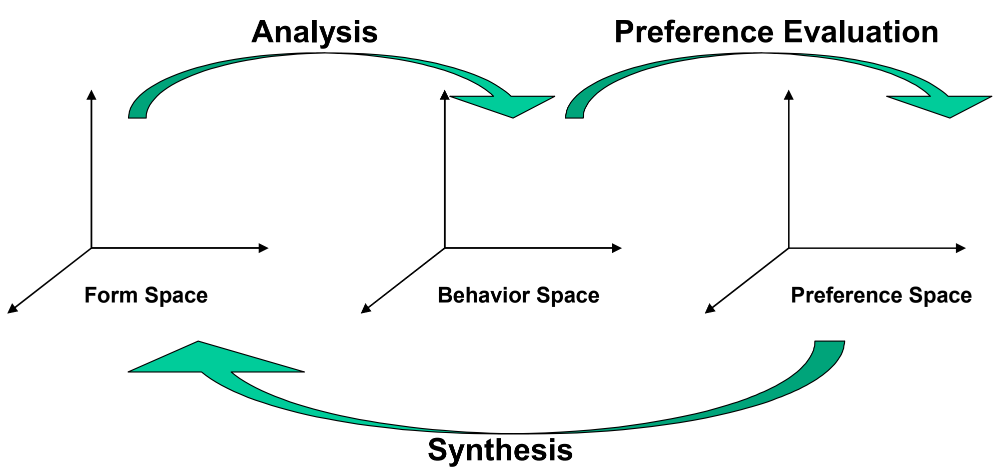


Figure 8-8 - Mappings in design processes

2. *Decomposition: Decomposition* is a transformation of product information that involves dividing the entity set into groups of entities that are designed independently. Partitioning of a design space segments the design space into sub-spaces. It also reduces the complexity of design problem. Ideally speaking, there

may be relationships between entities from different groups. However, if the interactions are very small, then these interactions can be neglected. For example, a product model Π shown in Figure 8-9 consists of four entities $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$. The strength of interactions represented by solid lines is much greater than the interactions represented by dashed lines. Hence, this product model can be partitioned into two groups of entities $\{\varepsilon_1, \varepsilon_2\}$ and $\{\varepsilon_3, \varepsilon_4\}$. These entities can then be designed separately. Partitioning can also be carried out at the parameter level where a group of parameters are separated into separate groups that are considered separately. Decomposition is a special case of partitioning where there are no interactions between the entities from different groups.

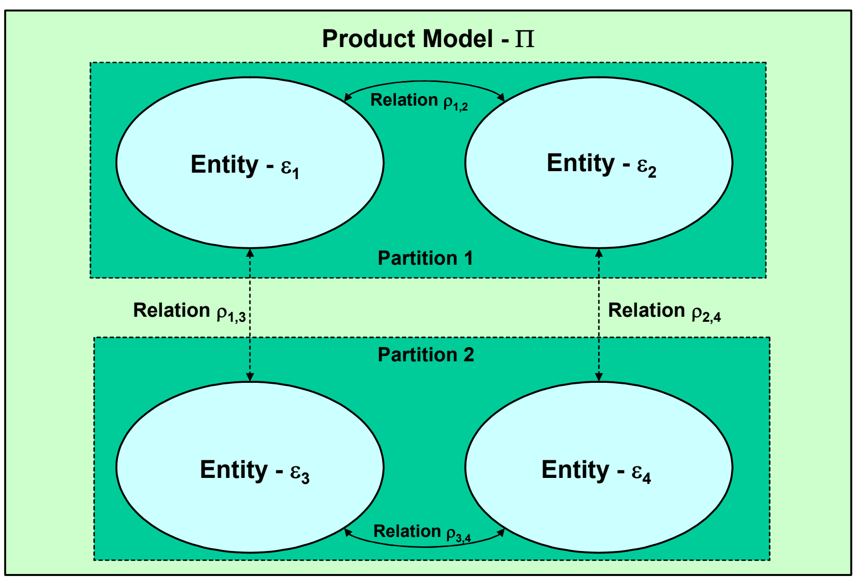


Figure 8-9 - Decomposition of entities in a product model

3. *Composition* refers to the synthesis of independently designed components of a system into a consistent whole. The primary challenge during composition is to consider coupling of phenomena due to interactions between components. From a decision standpoint, it is important to ensure that the decisions made by different

designers about different sub-systems of a previously partitioned system are consistent with each other. Composition is carried out during the system level synthesis after the sub-system level synthesis has been performed.

4. *Abstraction* refers to removal of detail from the product information, which is not important from the perspective of the design problem currently being solved. For example, during the structural analysis of a part, small features that do not contribute much to the overall strength are often ignored. This process simplifies the design process drastically. Abstraction is important for reducing the design space so that design space exploration is possible in the available time. For example, in a multi-scale design problem, the micro-scale and atomistic models contain many degrees of freedom that may not be required for the decision under consideration. Designers need to reduce the degrees of freedom in order to achieve the design objectives. At nanoscale analysis, the degree of freedom is determined by the arrangement of all the atoms. However, we don't need that much freedom for the design. There is a need to maintain the minimum set of design variables open in order to achieve a design objective. Abstraction is also important when there is a need to identify commonality between multiple systems. Further, in this dissertation, abstraction is used to simplify design patterns by ignoring couplings between microscale and macroscale simulation models.

5. *Refinement* refers to adding details to the product space. This can be carried out either by adding attributes to an entity that describe more details about the product under consideration, or by adding entities that are components of the higher level

entity. Refinement can be used to make the analysis process more accurate by adding important details that were not previously considered. This is shown in the previous chapters where design patterns are systematically made accurate by adding information about relationship between models – from independent Pattern P1 to sequential Pattern P2 and then to coupled Pattern P3. Refinement can also be carried out for individual simulation models by adding more details about the system. The information associated with refinement can be captured in a declarative form in refinement support problem template. The execution of this support problem it is carried out through a process used for refinement.

6. *Evaluation* refers to the process of determining how well a specific instantiation of an entity in the entity set complies with given criteria. The evaluation transformation is present in any design method. For example, in the Pahl and Beitz design method (Pahl and Beitz 1996) evaluation transformation is used to select one alternative from a set of alternatives that embody a function. Evaluation is based on determining the value of a metric quantifying the distance between *a)* desired point and *b)* the point corresponding to an alternative in the preference space. Sometimes, the distance is directly measured in the behavior space. But there, the implicit assumption is that the mapping between behavior space and preference space is linear. The distance between two achievable behavior points can be used as a selection criterion in concept selection methods. However, this distance needs to be evaluated on the preference spaces (and not on the behavior space). Many different selection methods are based on different methods, metrics

used for evaluation. In the compromise DSP, the metric for evaluation is based on the deviation of achieved values for goals from the target values.

7. *Decisions* refer to the reduction of entities in the entity sets based on some evaluation criteria. Evaluation and decision transformations generally go hand in hand. Designers can make rational decisions only based on the availability of consistent metrics. From a decision-based design perspective, decisions are the most important information transformations. However, it is discussed at the end because this transformation has been well formalized in the literature on decision-based design. In the DSP Technique, two decisions – selection and compromise have been identified as the only two basic decisions in design. All decisions can be expressed as combinations of these basic decisions. Due to their importance in decision-based design, we present an information model to capture the decision related information in a computationally interpretable manner, in Section 8.2.2.

8.2.2 Schema for Modeling Decision Problems

In this section, we present the schema for Decision Support Problems. The topmost entity is a *DecisionProblem*. This decision problem contains all the declarative information related to a decision support problem. The decision problem consists of four important elements – design space, response space, problem constraints, and preferences. Design space is defined by all the design variables that can be controlled by designers. Design variables can be either real or discrete. Real design variables have a continuous range values they can assume. Response space is defined by all the parameters that constitute the behavior space. Parameters in the response space have targets associated with them. These targets are derived from the customer requirements through the mapping transformation. Both the design variables and response variables are special types of attributes, described in the schema for product model in Section 8.1.2. Analogous to the design variables, response variables can also be discrete or real. It is important to note that the relationship between design variables and response variables is not defined in the problem description, but is defined in the product specific information model. This separation of information is important for reusability.

The third element of the design problem is problem constraint, first two being design space and response space. A constraint part of problem definition captures only the constraints that are due to way in which problem is defined. Product specific constraints are not defined in this section. They are captured using the relationship part of the product model. Constraints can be of two types – equality and in-equality. The fourth component of the design problem representation is preference. The preference part of the information model captures how much a designer values different outcomes in a manner that can be mathematically evaluated. These preferences can be captured in different mathematical forms – Archimedean, pre-emptive (Struble, Bascaran et al. 1989; Mistree, Hughes et al. 1993), or using utility functions (Seepersad 2001). In the Archimedean formulation, different goals are assigned weights and the overall objective function value is evaluated by taking the weighted sum of individual goals. In the pre-emptive formulation, different levels of objective functions are defined. After the higher levels are satisfied, designers can proceed to satisfy the next level of objective function. Using the utility-based preference representation, the preference values can be defined to vary with the value of each goal. Since multiple goals can be defined, the information model supports multi-objective decision making.

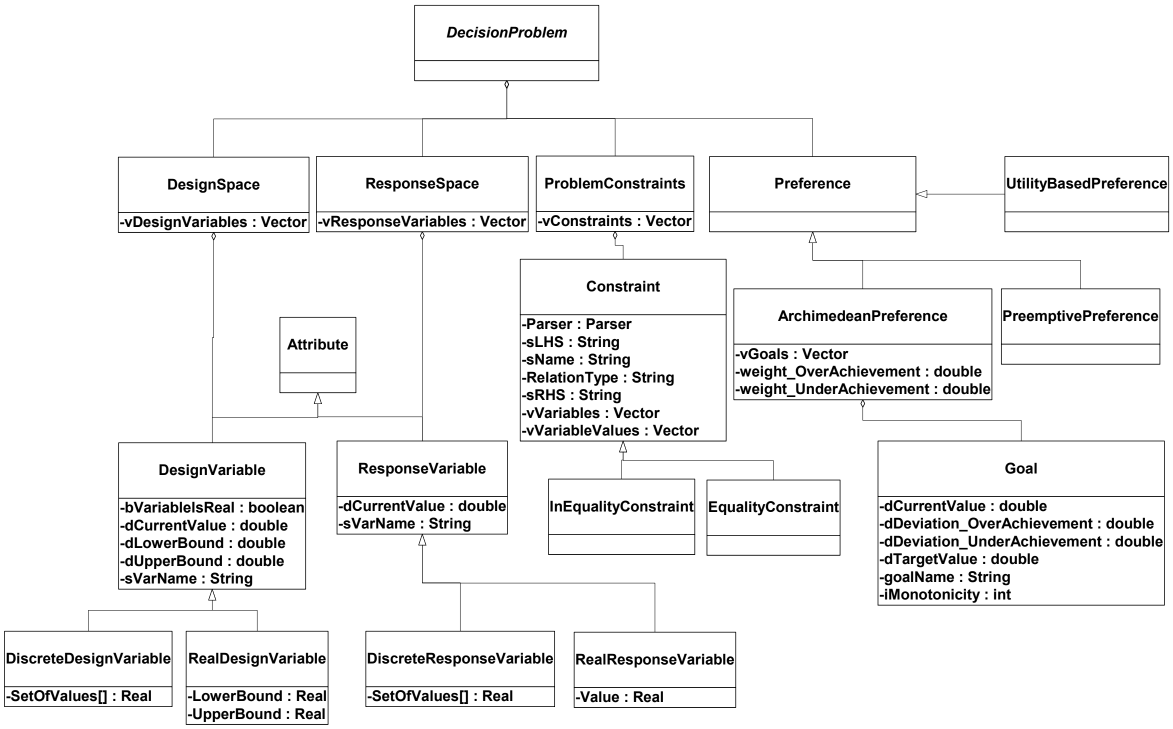


Figure 8-10 - Schema for capturing decision problem related information

8.2.3 Examples of Decision Support Problems

An example of a simple compromise DSP template is shown in Table 8-1. The key elements of the information captured in cDSP include design variables and their ranges, constraints, goals, relationship between attributes, and preferences. This is an un-instantiated template because there is no product specific data in the template. The cDSP can be instantiated by getting information from the product model and by specifying problem related goals, constraints, and preferences.

Table 8-1 - Generic decision formulation using cDSP

Given
Relationships between attributes (simulation codes)
Design variables, ranges
Targets for goals
Find
Design variables
Deviation variables
Satisfy
Constraints
Goals
Minimize
Deviation function (which is a function of satisfaction of goals, and their targets)

Table 8-2 - Decision formulation for robust design

Given
Relationships between attributes (simulation codes)
Design variables, ranges
Target values and variance for goals
Preferences for goals and their variance
Find
Design variables
Deviation variables
Satisfy
Constraints
Goals -
a. Achieving mean to target
b. Minimization of variance
Minimize
Deviation function (which is a function of satisfaction of a) mean value of goals, and their targets, and b) variance and their targets)

An adapted form of this compromise DSP for robust design is shown in Table 8-2. The difference between the two compromise DSP templates is the manner in which the goals are formulated. In the robust design case, each goal is associated with two sub-goals – *a*) achievement of target performance and *b)* minimization of variance in the achieved value due to variance in the noise variables (Chen, Allen et al. 1996; Chen, Allen et al. 1997). The compromise DSP for robust design is an extension of the information generic decision formulation presented in Table 8-1, and can be extended using the inheritance concept from object-oriented programming. An object-oriented information model for modeling decisions is discussed next.

8.3 PROCESS MODEL – ACTIVITIES SUPPORTING TRANSFORMATIONS

Design processes are defined at two levels – at the transformation level and at the activity level where activities that support execution of transformations are modeled. In this section, we only deal with activity level. The processes are modeled in this dissertation as a network of activities with information flowing from one activity to another. These activities are computational tasks in the context of simulation-based design. Each activity has a set of inputs and outputs. The manner in which outputs of one activity are modified to serve as inputs to another activity is referred to as an *interface* between the two activities. Interfaces are used to perform functions such as formatting or parsing of data. The process model also captures the information about the sequence in which activities are executed.

Two examples of processes for executing a decision problem are shown in Figure 8-11 and Figure 8-12 respectively. In the first example, an exhaustive search is used for finding the best point in the design space given various constraints and goals. The process involves four activities that include selection of a point from the design space, evaluation of relationships, evaluation of objective function, and updating the objective function. The execution of tasks involves utilization of information from problem definition. This process is used in the second example as a sub-process (see activity 4 in Figure 8-12). Other three tasks are used in this process include design of experiments to select points in the design space, execution of simulation code at those points, and response surface modeling. These activities are added to reduce the computational cost involved in executing the simulation codes. This example shows the need for reusability of design processes in higher level processes through composability.

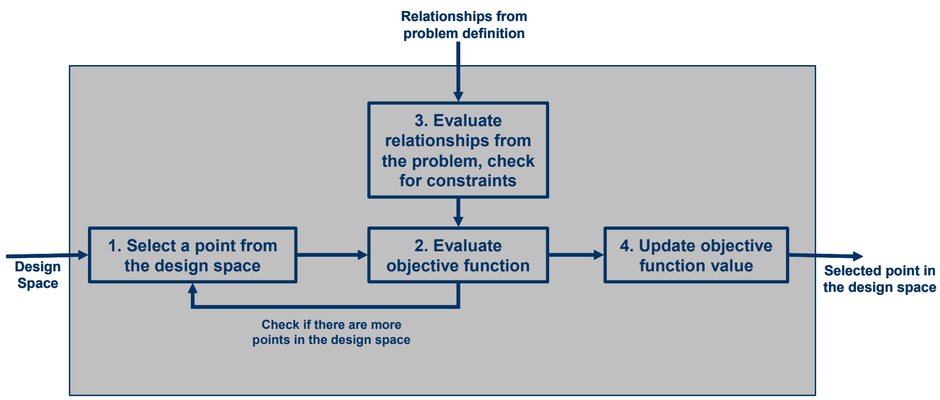


Figure 8-11 - Exhaustive search process for decision execution

The information model used for capturing process information is shown in Figure 8-13. The highest level element in the information model is a *Process*. The process is composed of two types of elements – basic and composite process elements. The basic process element can be directly executed on the computer whereas composite process element is a composition of other process elements. A composite process element has a process graph associated with it, which captures the information about its execution sequence. Two process elements with information flow between them are associated with an interface. The interface defines the outputs of one process element and the inputs of other process element, and also defines how they are mapped to each other in the *MappingMechanism* object.

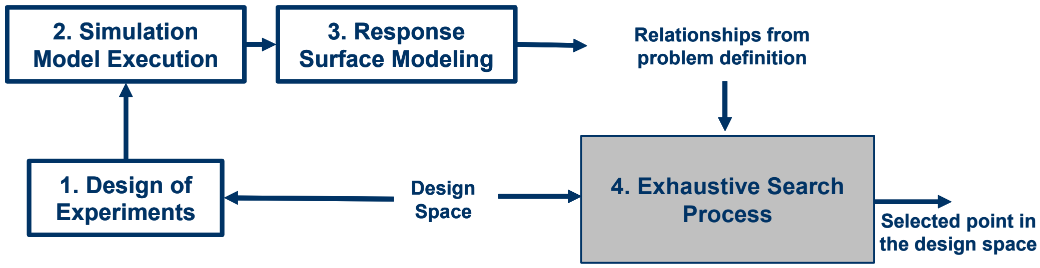


Figure 8-12 - Meta-modeling based process for decision execution

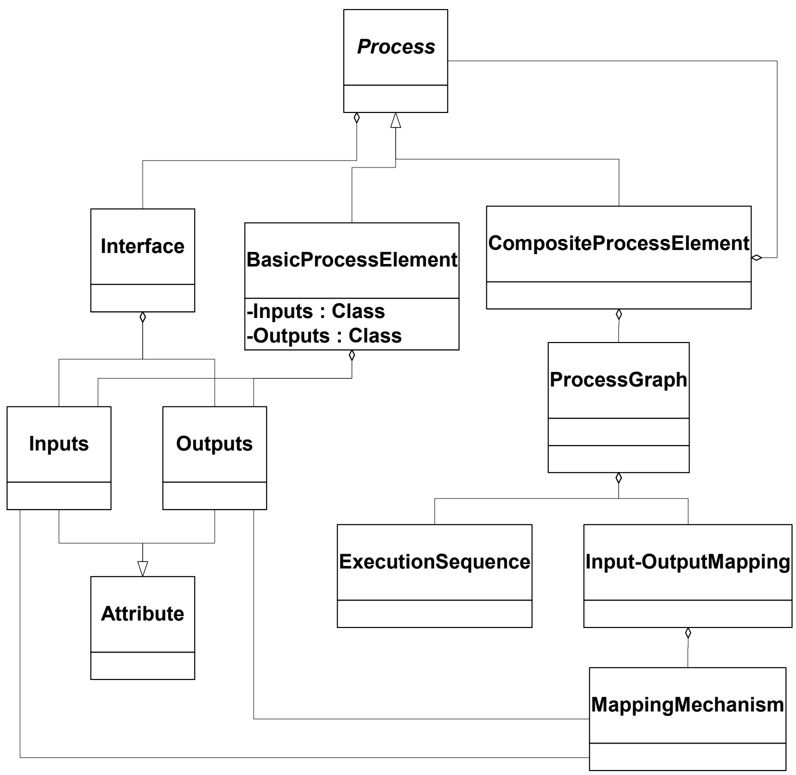


Figure 8-13 - Schema for capturing process information

We recognize that the information models presented in this chapter to support the 3-P approach are relatively simple and defined at a high level of abstraction. However, this is not a limitation of the overall approach presented. Comprehensive information models such as the core product model can be used to enrich the semantics of the information models presented.

8.4 Implementation of the Proposed Modeling Approach in ModelCenter

The 3-P information strategy is based on the separation of information related to products, processes, and design problems. Such a separation of information allows utilization of different design processes for solving a design problem, which facilitates meta-design. The 3-P strategy also supports utilization of the same design process for designing different products. As mentioned previously, the current simulation-based design frameworks capture product, process and problem related information in a strongly coupled manner and hence, restrict meta-design. In order to validate the 3-P information modeling strategy in the context of currently available simulation-based design frameworks, we consider only one aspect of the 3-P approach – reusability of same design process for different product to be designed. The use of different processes for the same product is not considered in this dissertation and is a potential for future work. Similarly, the combination of different types of design problems for same product is not considered.

In order to illustrate the use of same design process for different products, we consider a simple example, involving the design of two commonly employed mechanical components, namely, a pressure vessel and a spring, pictured in Figure 8-14. While both of these products can be described in terms of the geometric constraints, describing their form, and mechanical relations describing their function, they are nevertheless fundamentally different – with regard to the design parameters describing form, function, and behavior. Hence, computational design processes are problem specific and cannot be directly leveraged from one problem to another.

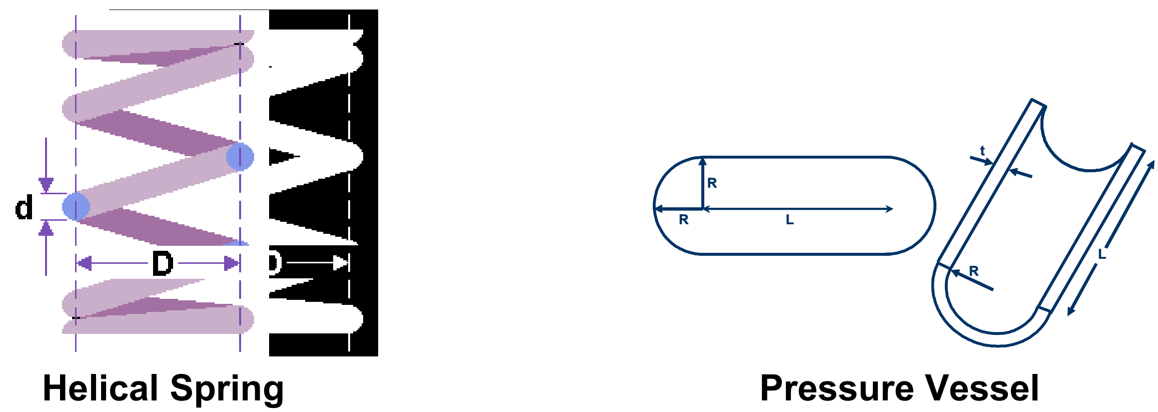


Figure 8-14 - Helical spring and pressure vessel

The information modeling approach presented in this section centers on the concept of modular separation of product and process specific information. In order to facilitate reusability of design processes across design problems, relevant information used to characterize them is segmented into three layers, as shown in Figure 8-15. Each of these layers (i.e., the *product information* layer, the *declarative process* layer, and the *execution* layer) is discussed in detail in this section.

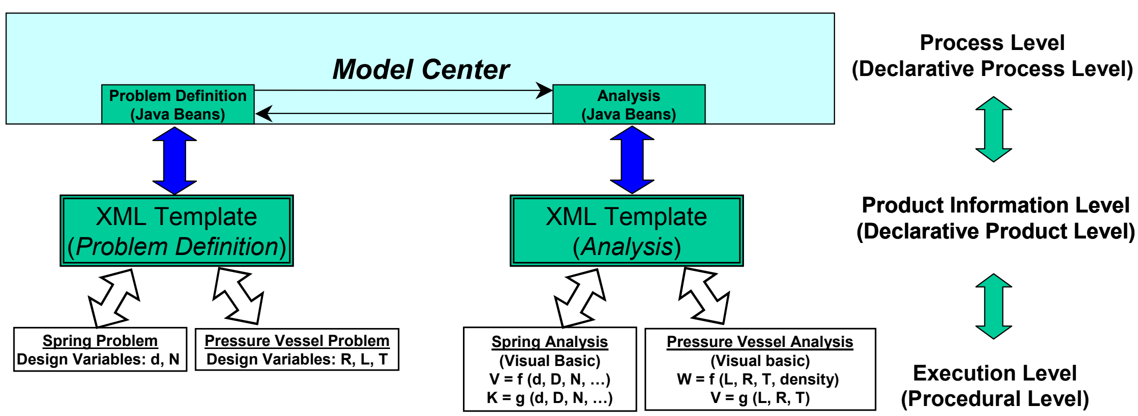


Figure 8-15 - Architecture of process modeling framework

Product Information Level (Declarative Product Level)

In the layer corresponding to the product information level, only information, specific to the product being designed, is captured. Since this information is treated in a standardized manner, it can be used by different design processes. For example, the information associated with the design of either the spring or the pressure vessel can be categorized as being comprised of design variables, responses, parameters, constraints, goals, preferences, objectives, or analyses. This is illustrated in their respective compromise DSP formulations given in Table 8-3.

Table 8-3 - Compromise DSP formulations for pressure vessel and spring design

Pressure Vessel Design	*Spring Design*
Given *Strength (S_t), Pressure (P), Density(ρ)* *Some helpful relations:* Volume, V = $\pi\left[\frac{4}{3}R^3 + R^2L\right]$ Weight, W = $\pi\rho\left[\frac{4}{3}(R+T)^3 + (R+T)^2L - (\frac{4}{3}R^3 + R^2L)\right]$ **Find** *System variables:* Radius (R) Length (L) Thickness (T) *Values of Deviation Variables:* *d1- (for weight goal)* *d2- (for volume goal)* **Satisfy** *System constraints:* $S_t - \left(\frac{PR}{T}\right) \ge 0$ $R - 5T \ge 0$ $(40 - R - T) \ge 0$ $(150 - L - 2R - 2T) \ge 0$ *System Goals (Normalized):* $d^-_{Volume} = 1 - \frac{V_{achieved}}{V_{target}}$ $d^-_{Weight} = 1 - \frac{W_{target}}{W_{achieved}}$ *Bounds on System Variables:* $0.1 \le R \le 36$ $0.1 \le L \le 140$ $0.5 \le T \le 6$ **Minimize** *Deviation Function:* $Z = w_1 d_1^- + w_2 d_2^-$	**Given** *Assumptions:* *Some helpful relations:* Deflection of spring: $\delta = \frac{8FD^3N}{d^4G}$ Solid height of spring: H=Nd, H≤0.5 in Stiffness of spring: $k = \frac{d^4G}{8D^3N}$ Volume of spring: $V = 0.25\pi^2 Dd^2(N+2)$ **Find** *System variables:* Wire diameter (d), Number of coils (N) *Values of Deviation Variables: d+, d- for goals* **Satisfy** *System constraints:* $6.957x10^{-6}\frac{N}{d^4} \ge 1.1$ $Nd \le 0.5$ *System Goals (Normalized):* $53345.5\frac{d^4}{N} + d_1^- - d_1^+ = 1$ $0.0191\frac{1}{d^2(N+2)} - d_2^- + d_2^+ = 1$ *Bounds on System Variables:* $N \ge 3.5$ $0.059 \le d \le 0.09$ **Minimize** *Deviation Function:* $Z = w_1 d_1^- + w_2 d_2^+$

We note that the two problems are quite different and exhibit dissimilar variables and relationships among them. The goals and constraints are also different. However, although the product specific information used in each formulation is different, the inherent structure according to which this information is used remains the same. Hence, it is possible to standardize the structure of information so that the creation of generic process elements becomes possible. The product information corresponding to these generic process elements for both the pressure vessel design and the spring design example are provided in Figure 8-16.

cDSP “Chips”	Pressure Vessel		Spring	
Constraints	Stress: $\frac{PR}{T} - S_t \leq 0$ Thickness: $5T - R \leq 0$ Radius: $R + T - 40 \leq 0$ Length: $L + 2R + 2T - 150 \leq 0$		Minimum Deflection: $\delta = \frac{8FD^3N}{d^4G} \geq 1.1$ Maximum Solid Height: $H = N \bullet d \leq 0.5$	
Variables	Radius, Length, Thickness		Number of Coils, Wire Diameter	
Parameters	Density, Strength, Pressure		Applied Force, Coil Diameter, Shear Modulus	
Goals	Volume Target = 500000 m^3 Weight Target = 300 kg		Stiffness Target = lbf/in Volume Target = in^3	
Preferences	Volume Weighting Factor = 0.5 Weight Weighting Factor = 0.5		Stiffness Weighting Factor = 0.5 Volume Weighting Factor = 0.5	
Objective	**Maximize** Volume $V(R,L) = \pi\left[\frac{4}{3}R^3 + R^2L\right]$ **Minimize** Weight $W(R,T,L) = \pi\rho\left[\frac{4}{3}(R+T)^3 + (R+T)^2L - \left(\frac{4}{3}R^3 + R^2L\right)\right]$		**Maximize** Stiffness $k = \frac{d^4G}{8D^3N}$ **Minimize** Volume $V = \frac{1}{4}\pi^2Dd^2(N+2)$	
Analysis	**Inputs**	**Outputs**	**Inputs**	**Outputs**
	Radius, Length, Thickness, Density, Strength, Pressure	Volume, Weight	Wire Diameter, Coil Diameter, Shear Modulus, Number of Coils	Stiffness, Volume
Driver	**Optimization Algorithm** – Exhaustive Search, SQP, etc.		**Optimization Algorithm** – Exhaustive Search, SQP, etc.	
Response	**Design Variable Values** – Radius, Length, Thickness **Objective Function Value** - Z		**Design Variable Values** – Radius, Length, Thickness **Objective Function Value** - Z	

Figure 8-16 - Product information level (declarative product level) for pressure vessel and spring design

The process modeling technique proposed in this dissertation is analogous to architecture of a printed wiring board with a number of electronic components, such as those shown in Figure 8-17. The *wiring* corresponds to the flow of information in a process and the declarative process specific information is thus “hardwired”. The *chips* plugged into the board define the manner in which the information is actually processed. Consequently, these chips correspond to the declarative (product specific) information, discussed in this section. A prime benefit is that the resulting reusability extends to both the chips and the board independently. Since procedural elements of information transformations are captured in a template form that is independent of the declarative aspects (i.e., the specific information considered), all aspects of information transformations from the components to the underlying interactions (represented by the "chips" and "wiring" in Figure 8-17, respectively) become modular. Both re-usability and reconfigure-ability are thus achieved.

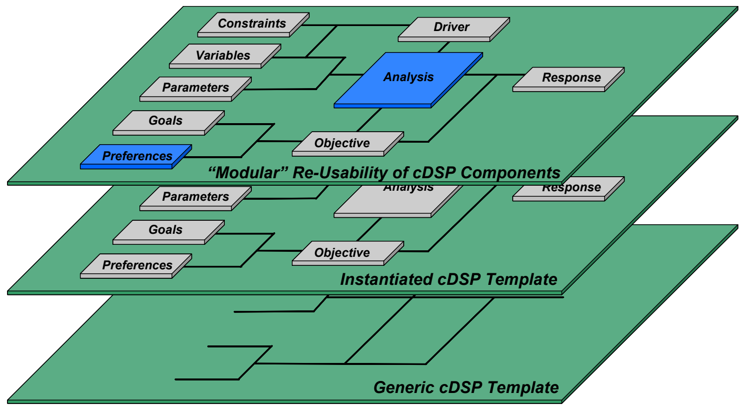


Figure 8-17 - Archival, documentation, and re-use of design process building blocks

Currently, we standardize the structure of product information according to a set of XML schemas. XML offers a convenient and standardized means of capturing information at the product information level and ensures that problem specific *declarative* information can be reused in different processes. For the simple example problem of designing both a pressure vessel and a helical spring through the use of a common template, the product information is stored in four XML templates: the *problem definition template*, the *constraints template*, the *goals and preferences template*, and the *analysis code template*. These templates, discussed next, correspond to the declarative product information "hidden" (or embedded) within the compromise DSP formulation shown in Table 8-3. Note that the information captured in these schemas is not a direct implementation of the 3-P information models presented in Figure 8-5, Figure 8-10, and

Figure 8-13. These simple schemas are used just for illustration of the concept of separation of declarative and procedural information.

Variables and Parameters Definition Template

The template for defining design variables and parameters includes the following information about design variables: a) Design Variable Name b) Type c) Unit d) Value e) Lower Bound and Upper Bound. For the purposes of this section, all parameters are defined with equal lower and upper bounds. The XML schema representation associated with the problem definition template is shown in Figure 8-18.

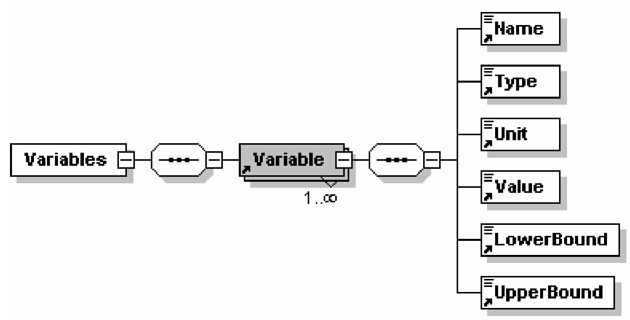


Figure 8-18 – XML schema representation for variable definition

Constraints Definition Template

The constraints definition template includes information about various constraints on the system. The constraints are associated with a name and a string representing required mathematical operations. The XML schema representation associated with the constraints definition template is provided in Figure 8-19.

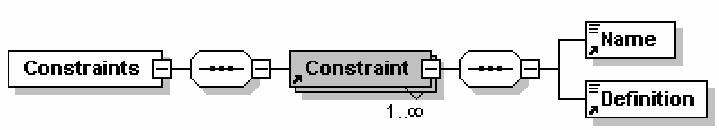


Figure 8-19 – XML schema representation for constraints

Goals and Preferences Definition Template

In this template, information about design goals and designer preferences regarding the satisfaction of these goals is captured. The goals are formulated with target values for system responses. Preferences are associated with the various goals included in the compromise DSP formulation. Here, these preferences are modeled as weights on the deviation variables. The entities associated with such goals are: a) Name b) Weight c) Target and d) Monotonicity, where Monotonicity captures information regarding whether the goal is to be maximized, minimized, or matched as closely as possible. The XML schema associated with the goals and preference definition templates is shown in Figure 8-20.

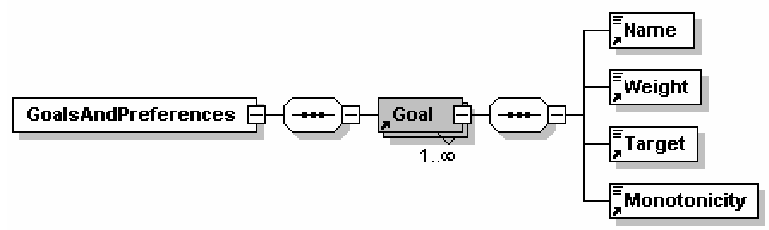


Figure 8-20 – XML schema representation for goals and preferences

Analysis Code Template

The analysis code is used to evaluate the system response to changes in design variables. The information associated with the analysis code template includes a) Inputs, which consist of Name, Type, Unit, and Value, b) Outputs, which consist of: Name, Type, Unit, and Value and c) Execution. The "Execute" field captures the software application that needs to be invoked in order to obtain the desired system response. The XML schema associated with the analysis code template is also shown in Figure 8-21.

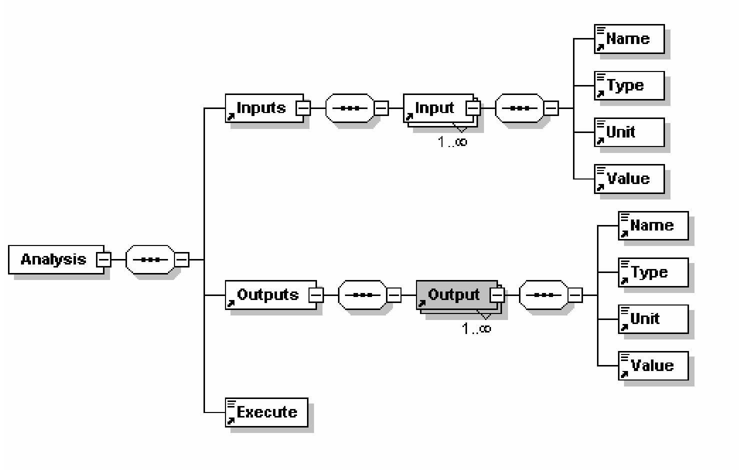


Figure 8-21 – XML schema representation for analysis model

Process Level (Declarative Process Level)

In the layer, corresponding to the Process Level, (1) required information transformations are identified and (2) required information flows are specified in accordance. In order to ensure complete modularity of information transformation templates, information flows are separated from information content. Effectively a clear distinction is made between *declarative* and *procedural* information content. In other words, we capture only the mechanics of information transfer at this level, while problem specific information is defined separately at the declarative level. This results in a process map that remains the same irrespective of the application in which the process is used. Information content is thus effectively batched, according to the structure of the overarching template.

A simple example of a generic process map for the design of either a spring or a pressure vessel using the compromise DSP construct is given in Figure 8-22. The elements of this generic process include problem definition, analysis, constraint evaluation, goal evaluation, and an optimization routine. Each of these entities interacts with the product information layer through the product information templates. The information flows between these entities are generic and independent of the product being designed. For example, the flow of information between the analysis module and constraints evaluation include the problem name, an array of input names (i.e., design variables) and an array of input values. The actual input names and values are dependent on the problem and are extracted from the variables and parameters definition template.

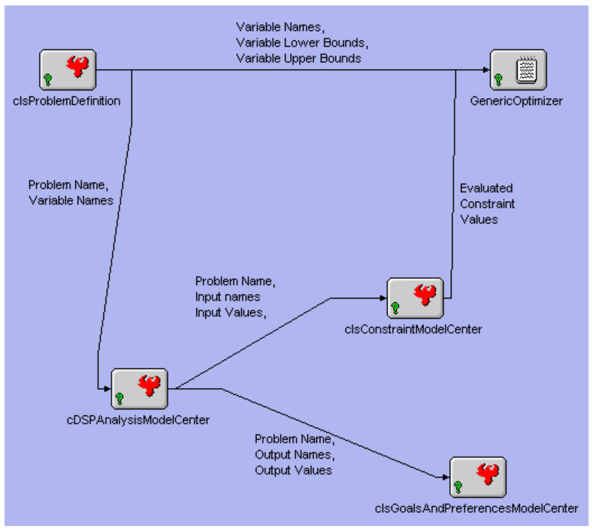


Figure 8-22 - Process map for design of spring / pressure vessel

The implementation of the declarative process level relies on the use of ModelCenter® (Phoenix Integration Inc. 2004), developed by Phoenix Integration Inc. ModelCenter® allows for modeling design processes in terms of various simulation codes and the required information flows connecting them. Associated with each entity in this process are a set of JavaBeans that parse required information from appropriate XML files at the product information level and subsequently make this information available for processing in ModelCenter®. These Process elements are mapped to each other for a specific problem, in a manner that reflects the underlying (batched) information flows required by the generic templates. This mapping remains the same irrespective of the design problem in which the process is used. For example, the information flows and mappings relevant for the solution of a compromise DSP, will remain the same, whether the product being designed is a pressure vessel or a spring.

Execution Level (Procedural Level)

The details of code execution are captured in the Execution Level layer. This level is specific to the design problem for which the process is used. Execution level codes interface only with the declarative problem formulation level. Thus, there is no direct link between the process specification level and the execution level. This architecture preserves the modularity of the design processes being modeled. For the design of the pressure vessel and the spring, the execution level codes (i.e., the analysis codes simulating the behavior of both the spring and the pressure vessel) have been written in Visual Basic, although any other model wrapped as a ModelCenter® component could also be used in the current instantiation of this modeling effort.

The results obtained for the pressure vessel and spring design using the generic process, pictured in Figure 8-22, are summarized in Table 8-4 and Table 8-5, respectively. These results have been verified and validated with exhaustive searches, based on more traditional problem formulations.

Table 8-4 - Results for pressure vessel problem

Design Variable	*Value*
Radius (R)	10 mm
Length (L)	80 mm
Thickness (T)	3.5 mm
Objective function (Z)	0.497

Table 8-5 - Results for spring design problem

Design Variable	*Value*
Coil Diameter (d)	0.059 in
Number of Coils (N)	3.5
Objective function (Z)	0.655

8.5 Role of Chapter 8 in This Dissertation

In this chapter, we present details for the implementation of the 3-P strategy. These include information modeling approaches and supporting schemas for products, problems, and processes. A part of the proposed strategy is validated via instantiation in an existing simulation-based design framework - ModelCenter. In the following chapter (Chapter 9), the methods and metrics developed throughout the dissertation are validated through a multiscale, multifunctional materials and product design example.

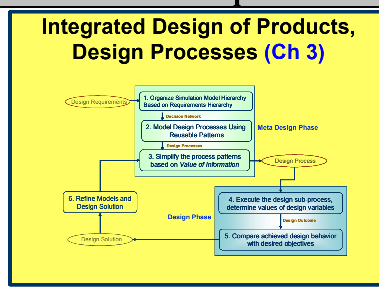
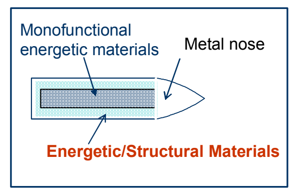
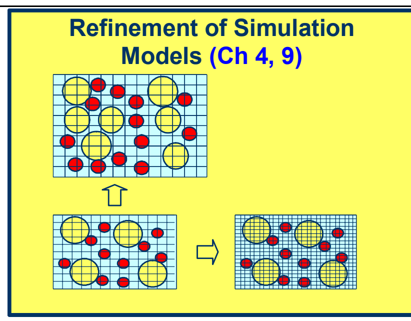
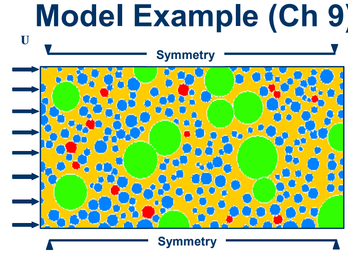
Chapter 9 Designing the Design Processes for Multi-scale, Multi-functional Materials Design

9.1 Context – Validation of the proposed design framework

The objective in this chapter is to validate two components of the design framework – *a)* method for integrated design of products and design processes (developed in Chapter 3), and *b)* value of information based metrics for model refinement and design process simplification (developed in Chapter 4 and Chapter 5), using a comprehensive design example. The objective of validation of the framework components is accomplished by selecting a design problem that involves design of multifunctional materials, products and design processes. It is a reasonably complex problem involving design at multiple scales and functionalities. Results from the example presented in this chapter are used for answering research questions RQ1 and RQ2 in the dissertation. These two components of the framework validated in this chapter and the associated framework requirements are highlighted in Table 9-1. This table is a subset of Table 1-6 presented in Chapter 1 to map the requirements, framework components, and validation examples.

In this chapter, we start with an introduction of a general multiscale materials design problem in Section 9.2. The section contains a requirements statement for a multifunctional material for energetic and structural applications. In order to design the material, a number of analysis models are required at various scales. These different scales of simulation models are discussed in the section. Following the discussion of the general materials design problem, a specific design subproblem is formulated in Section 9.3, and is addressed throughout this chapter. The problem consists of design of materials in conjunction with the products and the underlying design processes. Specific details of three simulation models used to design materials and products are provided in Section 9.4. The details of information flows between these models are also discussed.

Table 9-1 – Framework requirements and associated components validated in Chapter 9

Framework Requirements	Components of the Framework Developed to Address the Requirements	Validation Examples
1) A method for integrated design of products and design processes		**Materials-Product design example (Ch 9)**  ***Purpose:*** To validate the method for integrated design of products and design processes
5) Support for evolving simulation models		**Particle Shock Simulation Model Example (Ch 9)**  ***Purpose:*** To validate the use of value-of-information based metrics for simulation model refinement

The application of design method proposed in Chapter 3 to solve the design problem is presented in Section 9.5. All the steps of the design method except model refinement are discussed in Section 9.5. In order to manage the complexity of the problem, model refinement is taken up separately in Section 9.6, where a specific simulation model is refined for a design problem. In that section, the integrated nature of simulation model refinement and material decisions is explored. Verification and validation of the method are addressed in Section 9.7. Finally, the role of this chapter in the dissertation is discussed in Section 9.8. The hypotheses addressed in this chapter are highlighted in Figure 9-1. In this chapter, we address the validation of hypotheses H1.1, H1.2, H2.1, and H2.2.

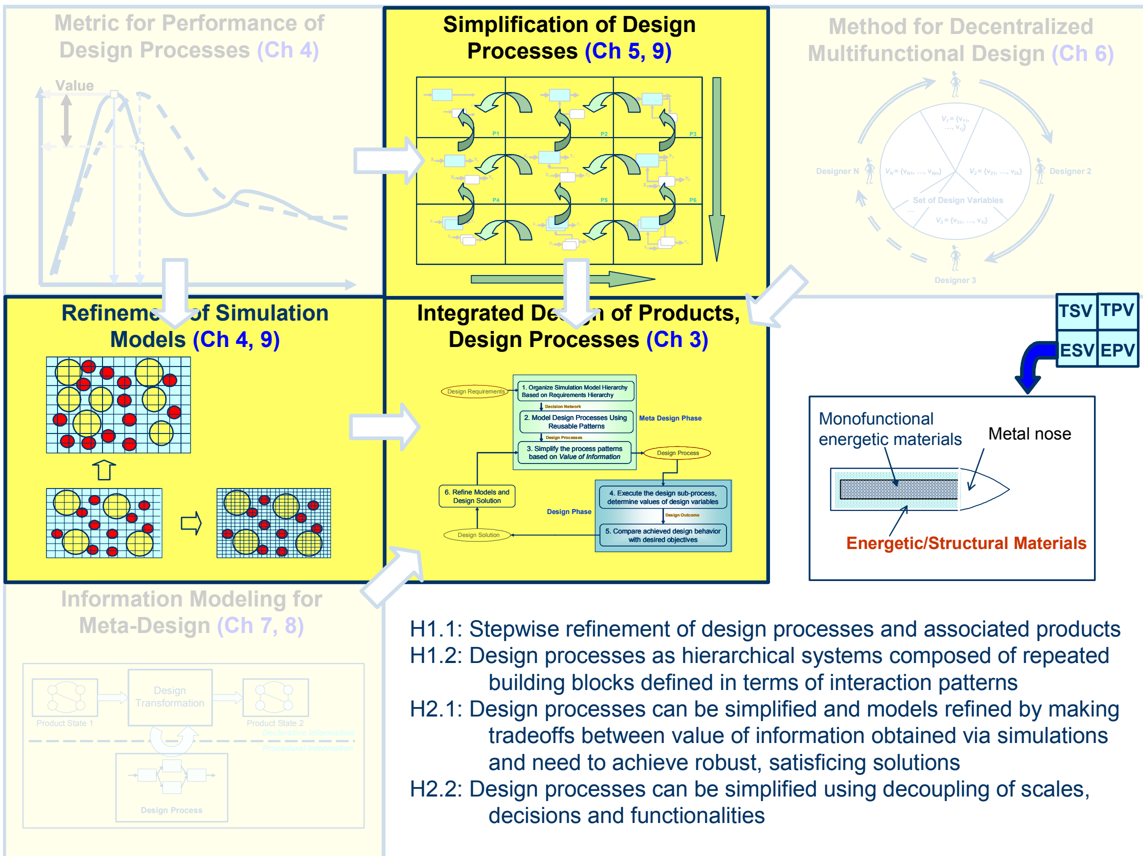


Figure 9-1 – Hypotheses addressed in Chapter 9

9.2 *Illustration of the Material design scenario*

In materials design, multi-time-scale and multi-length-scale analyses and simulations are performed to tailor a material for specific needs (see Figure 9-2). One such class of tailored materials is Multi-functional Energetic Structural Materials (MESMs). MESMs, which may be composed of Reactive Powder Metal Mixtures (RPMMs), are unique in that the components serve the dual purpose of providing both energetic fuel and structural integrity to a reactive system. We are working on an AFOSR sponsored MURI on design of MESMs for a multifunctional application, where the design problem statement is as follows -

"Design multi-functional materials spanning micro and nano-scale structural architectures that can provide dual functions of strength and energy release. These materials can be used to design structures that support and house low load-bearing mono-functional energetic materials and can withstand loads that result from high striking velocity impact of selected solids. During the impact event, the structure should not fail as result of the impact loads or initiate chemical reactions while striking the solid of interest. The solid of interest can fail or yield due to the impact or penetration mechanics. The structure that is made of multi-functional materials of interest must predictably function after the impact event."

Both the energetic and structural properties are highly dependent upon particle properties such as size, distribution, and volume fractions of the particles, as well as, the composition of filler materials. In this example, numerical values of these properties are to be determined in order to achieve the desired performance of the resulting material. These numerical values are based on the material behavior predicted using simulation models at various scales. Some of the important simulation models available in the literature at different scales include:

1. *Quantum mechanics models:* Quantum scale models are used to determine equation of state properties of the individual materials and likelihood of reaction initiation between reactive components. These ab-initio models only require environmental

properties such as pressure and temperature as inputs as the atomic properties are fundamentally derived. The equation of state results, particularly in the Hugoniot form, are used in the mesoscale discrete particle models to determine constitutive behavior of the individual components. Evaluation of the transition states and energies relative to the interaction of reactive components can be processed to determine the likelihood of a reaction to initiation. These probabilities can be used in the mesoscale discrete particle models. Quantum scale models are also used to determine the parameters in potentials used in molecular dynamics models.

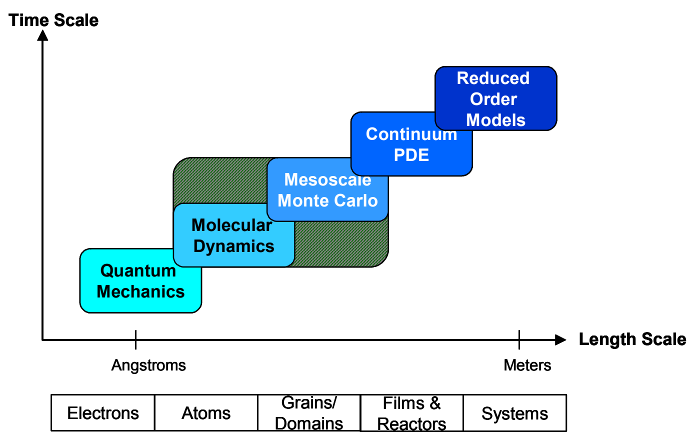


Figure 9-2 – Modeling at multiple length and time scales for materials design

2. *Molecular dynamics models:* Molecular dynamics (MD) models can also be used to investigate equation of state properties and probability of reaction initiation, but at larger scales. While quantum scale models are limited to tens or hundreds of atoms, MD models can investigate the interactions of 100's of thousands to millions of atoms. The MD models can then be used more effectively to study shock waves through the atoms, size scale effects at reactive component interfaces, and nanoscale domains of the constituents. The MD models can also be compared and verified with

quantum scale models. The results of the MD models would be used in the meso-scale discrete particle models.

3. *Meso-scale models:* In the meso-scale model, the constituents are modeled as discrete particles in the nanometer to micron scale. With a given shock input, the model estimates reaction initiation and structural information within a volume element measuring in the tens to hundreds of microns in length. The randomly generated morphology is created based on statistical information such as volume fractions, size distributions, and nearest neighbor distributions. The structural information in the form of Hugoniot data is used in continuum models for systems model analysis. The reaction initiation aspects of the model can be used to determine the likelihood of reaction propagation and the pressure, temperature effects of the reaction.
4. *Continuum PDE models:* Achieving accurate continuum model for new materials is an important task for accurate performance estimation in a system scale model. Structural and reaction information generated from mesoscale model are interfaced with the continuum model. Experimental data are often required for validating analysis models at this scale. Constitutive models and reduced order models are implemented at this scale. The reduced order model in this MESMs analysis, is a non-equilibrium mixture model which predicts structural and reaction relations based on the 'rule of mixtures'.

These simulation models are common to most multiscale materials design problem. These simulation models are associated with complex information flows. Such an information flow between models at various scales is shown in Figure 9-3. With this general multiscale materials design problem in mind, we proceed to present the specific design problem considered in this dissertation for the validation of method for integrated design of products, and design processes.

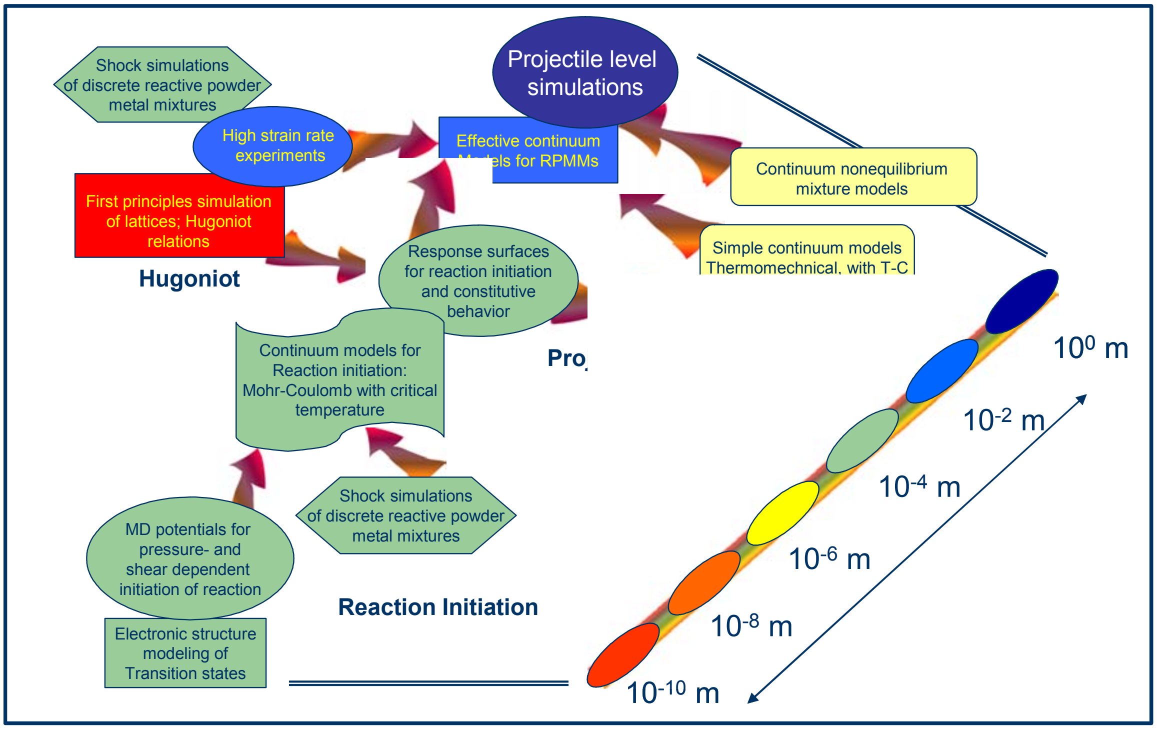


Figure 9-3 – Information flow between simulation models at various scales (figure courtesy David L. McDowell)

9.3 Problem: Integrated Design of Products, Materials, and Design Processes

In the previous section, a general materials design problem is presented. In this section, we present a specific design problem used in this chapter for the validation of constructs developed in the dissertation. The design problem involves designing a projectile and its material to satisfy multifunctional design objectives. The conceptual design of the projectile is fixed for the purpose of this chapter as shown in Figure 9-4, an outer shell of steel with an inner energetic structural material filling. It is assumed that the energetic structural material consists of a mixture of aluminum and iron oxide particles in epoxy binder. The objectives in the design problem are to satisfy the strength and energy release goals.

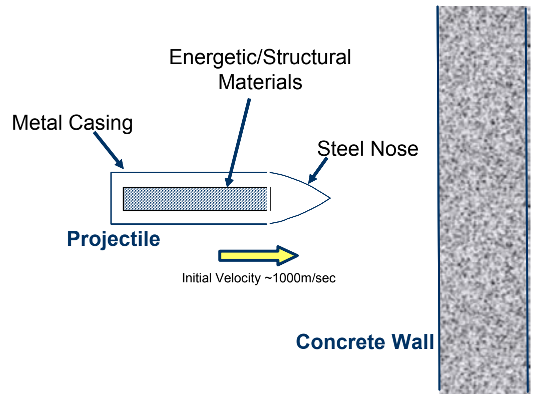


Figure 9-4 – Conceptual layout of the projectile impacting a concrete wall

In order to satisfy the objectives, the designers can make decisions about the following design variables with ranges as shown in Table 9-2.

Table 9-2 - Design variables and ranges for integrated materials-product design problem

Radius of aluminum particles: [0.0005 0.0015]mm
Radius of iron oxide particles: [0.0002 0.0010]mm
Radius of voids: [0.0002 0.0010]mm
Volume fraction of voids: [0.02 0.10]
Radius of the filling material in the projectile: [5 23]mm

All the other variables associated with the product and the materials are fixed. The preference for the objectives is expressed in terms of utility functions. The designers' preferences are associated with two functional characteristics – *a)* the strength of the projectile, and *b)* the reaction initiation and propagation properties. An indicator of strength is used in this dissertation to simplify the design problem. This indicator is the deformation achieved at a particular time by the projectile in a Taylor impact test (details are provided in Section 9.4.3). Similarly, an indicator for reaction properties of the material is used as a functional response. This indicator is the amount of reaction products (iron, in this material system) accumulated at a specified time after the shock starts. Utility functions are specified for both the responses and are shown in Figure 9-5. These utility functions are ideally generated based on the designers' preferences. In this specific problem, the utility functions are generated based on some assumed performance targets. The actual details of utility values would affect the final solution but not the application of design method.

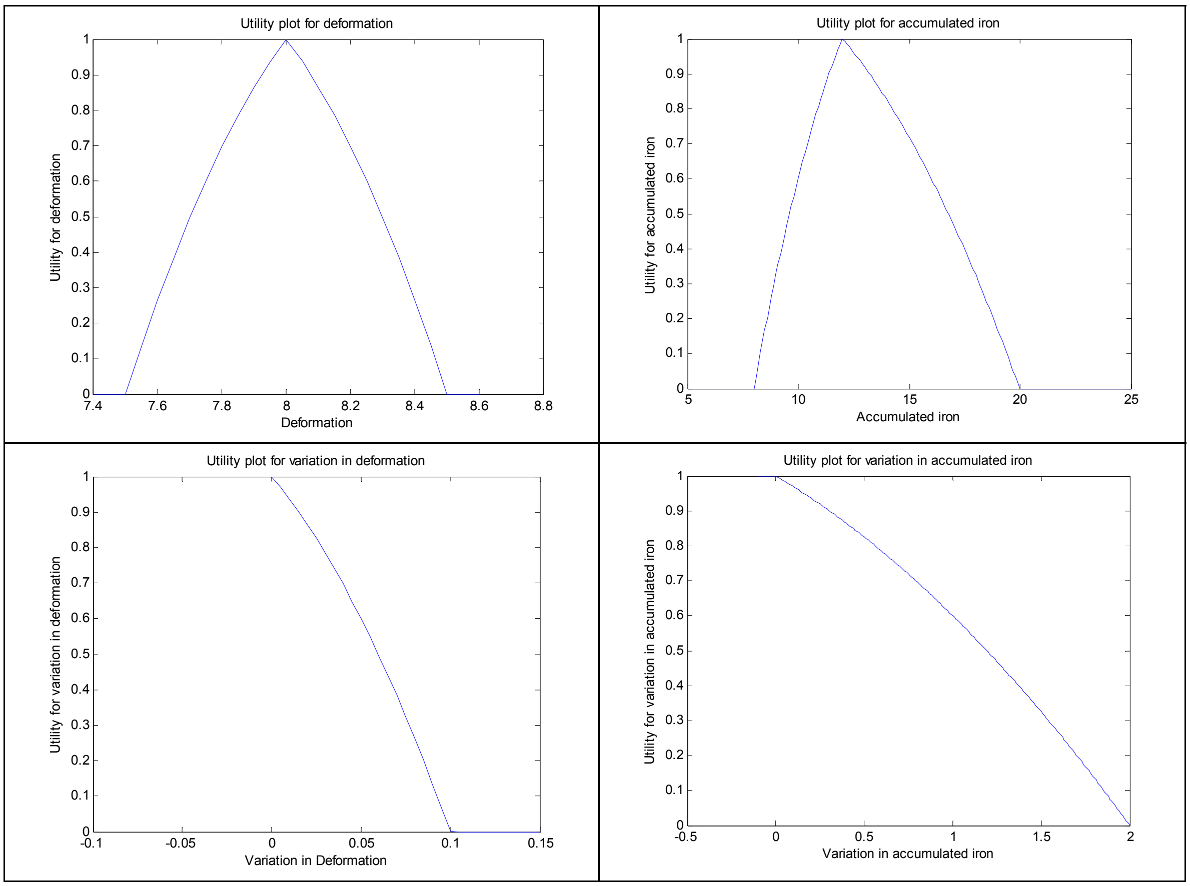


Figure 9-5 - Utility functions for deformation, accumulated reaction products and their variation

As is shown, the designer has a target matching preferences for both the responses (deformation and accumulated reaction products). In addition to the preferences for target achievement of the two response values, designers also have preferences for the deviation in these target values. These preferences for deviation in performance are important in the design problem due to the need for making decisions that are robust to variation in performance of the material. The variation in performance can be due to various factors such as inherent randomness in material properties, imprecision in simulation models, and the imprecision introduced due to simplification in design processes. The preferences for deviations are specified as monotonically decreasing utility functions as shown in Figure 9-5.

The compromise DSP representation of the overall (combined) decision about product and material is presented in Table 9-3. Note that the problem addresses a subset of the requirements statement presented in Section 9.2.

Table 9-3 – Decisions about product and material

Decisions about product and material
Given Simulation models at three levels *Preferences for deformation, accumulated iron, variation in deformation, and variation in accumulated iron* $U_{def} = \begin{cases} 1.4\left(\frac{def\text{-}def_{min}}{def_{avg}\text{-}def_{min}}\right) - 0.4\left(\frac{def\text{-}def_{min}}{def_{avg}\text{-}def_{min}}\right)^2 & if\,(def_{min} < def < def_{avg}) \\ 1 - 0.6\left(\frac{def\text{-}def_{avg}}{def_{max}\text{-}def_{avg}}\right) - 0.4\left(\frac{def\text{-}def_{avg}}{def_{max}\text{-}def_{avg}}\right)^2 & if\,(def_{avg} < def < def_{max}) \\ 0 & otherwise \end{cases}$ $def_{min}=7.5$, $def_{avg}=8.0$, $def_{max}=8.5$ 'def' refers to deformation

Decisions about product and material

$$U_{accFe} = \begin{cases} 1.4\left(\dfrac{accFe\text{-}accFe_{min}}{accFe_{avg}\text{-}accFe_{min}}\right) - 0.4\left(\dfrac{accFe\text{-}accFe_{min}}{accFe_{avg}\text{-}accFe_{min}}\right)^2 & if\,(accFe_{min} < accFe < accFe_{avg}) \\ 1 - 0.6\left(\dfrac{accFe\text{-}accFe_{avg}}{accFe_{max}\text{-}accFe_{avg}}\right) - 0.4\left(\dfrac{accFe\text{-}accFe_{avg}}{accFe_{max}\text{-}accFe_{avg}}\right)^2 & if\,(accFe_{avg} < accFe < accFe_{max}) \\ 0 & otherwise \end{cases}$$

$accFe_{min} = 8$, $accFe_{avg} = 12$, $accFe_{max} = 20$

'accFe' refers to accumulated Iron

$$U_{def\_Uncertain} = \begin{cases} 1 & if\,(def_{Uncertain} \le 0) \\ 1 - 0.6\left(\dfrac{def_{Uncertain}}{0.1}\right) - 0.4\left(\dfrac{def_{Uncertain}}{0.1}\right)^2 & if\,(0 < def_{Uncertain} < 0.1) \\ 0 & if\,(def_{Uncertain} \ge 0.1) \end{cases}$$

$$U_{accFe\_Uncertain} = \begin{cases} 1 & if\,(def_{Uncertain} \le 0) \\ 1 - 0.6\left(\dfrac{accFe_{Uncertain}}{2}\right) - 0.4\left(\dfrac{accFe_{Uncertain}}{2}\right)^2 & if\,(0 < def_{Uncertain} < 2) \\ 0 & if\,(def_{Uncertain} \ge 2) \end{cases}$$

Find

Values of design variables

Size of Aluminum Particles, Size of Fe2O3 particles, Size of Voids, Volume Fraction of Voids, Radius of filling material

Values of deviation functions

d^+_{def}, d^-_{def}

d^+_{accFe}, d^-_{accFe}

$d^+_{def\_Uncertain}, d^-_{def\_Uncertain}$

$d^+_{accFe\_Uncertain}, d^-_{accFe\_Uncertain}$

Satisfy

Bounds on design variables

- Material level variables

Size of Aluminum Particles = [0.0005 0.0015]mm

Size of Fe2O3 particles = [0.0002 0.0010]mm

Size of Voids = [0.0002 0.0010]mm

Volume Fraction of Voids = [0.02 0.10]

- Product level variable

Decisions about product and material
Radius of filling material = [5 23]mm *Utility goals* $U_{def} + d^-_{def} - d^+_{def} = 1$ $U_{accFe} + d^-_{accFe} - d^+_{accFe} = 1$ $U_{def\_Uncertain} + d^-_{def\_Uncertain} - d^+_{def\_Uncertain} = 1$ $U_{accFe\_Uncertain} + d^-_{accFe\_Uncertain} - d^+_{accFe\_Uncertain} = 1$ *Constraints on deviation variables* $d^-_{def}.d^+_{def} = 0$ $d^-_{accFe}.d^+_{accFe} = 0$ $d^-_{def\_Uncertain}.d^+_{def\_Uncertain} = 0$ $d^-_{accFe\_Uncertain}.d^+_{accFe\_Uncertain} = 0$ $d^-_{def}, d^+_{def}, d^-_{accFe}, d^+_{accFe}, d^-_{def\_Uncertain}, d^+_{def\_Uncertain} \geq 0$ $d^-_{accFe\_Uncertain}, d^+_{accFe\_Uncertain} \geq 0$ $k_1 + k_2 + k_3 + k_4 = 1$ **Minimize** $Z = k_1 d^-{}_{def} + k_2 d^-{}_{accFe} + k_3 U^-{}_{def\_Uncertain} + k_4 d^-{}_{accFe\_Uncertain}$

Since the design problem involves deciding on both the material design variables and the product design variables, the problem involves the integrated design of products and materials. The decisions about products and materials are coupled with each other because both decisions impact the deformation and reaction behaviors of the complete product-material system. Further, both material and product decisions require multiple simulation models that exchange information between each other. The simulation models are also coupled with each other. These couplings between decisions and the simulation models increase the complexity of the complete design problem. However, as emphasized throughout the dissertation, not all couplings are important from a decision making standpoint. Some couplings have a significant effect on the designers' decisions whereas others have only a small effect. Hence, this problem presents a need for determining which couplings can be eliminated from the design processes such that the information is generated and utilized in an efficient manner, thereby supporting effective decision-making. The problem is in line with the primary research question in this dissertation discussed in Section 1.1.5. The efficiency and effectiveness of utilization of information is one of the main aspects of designing design processes. Hence, in addition the problem of integrated decisions about products and materials shown in Table 9-3, the designers should also make decisions about the design processes in an integrated manner. We limit our discussion on designing design processes to determining which couplings (between decisions and simulation models) are important from a decision making standpoint. The integrated nature of these three types of decisions is shown in Figure 9-6. Having discussed the design problem, we now proceed to discuss the details of simulation models used for solving the design problem.

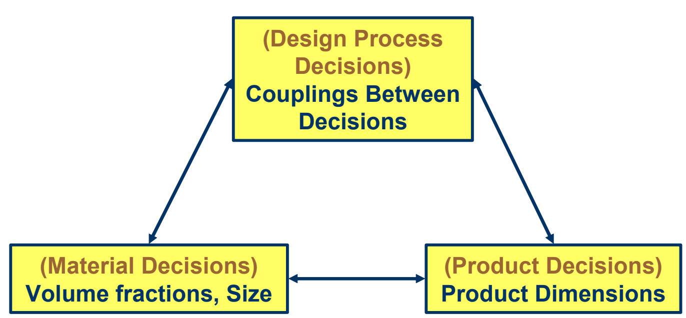


Figure 9-6 –Integrated design of products, materials and design processes

9.4 *Material and System Level Simulation Models Used in the Dissertation*

In Section 9.2, general types of models for simulating the behavior of material are discussed. However, in order to solve the design problem presented in Section 9.3, we rely on three models – particle-level shock simulation (micro-level), non-equilibrium mixture theory model (continuum level), and projectile level model (system level) The three simulation models at different scales are shown in Figure 9-7. The details of these three models are discussed in Sections 9.4.1, 9.4.2, and 9.4.3 respectively. flow of information between these three models is discussed in 9.4.4.

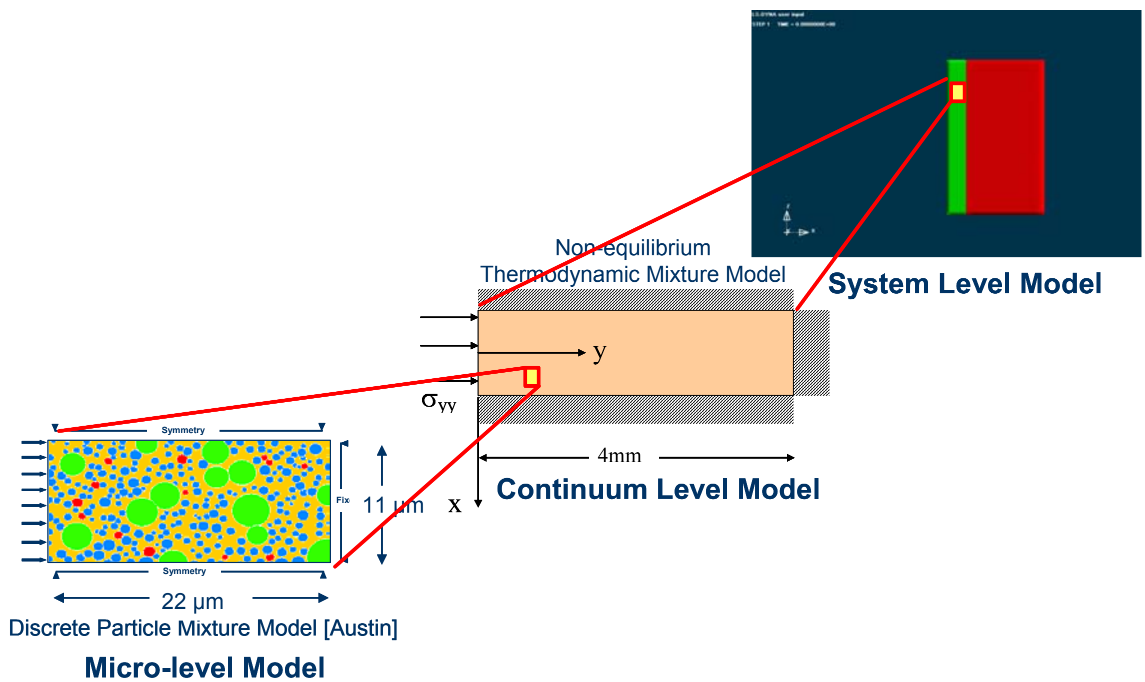


Figure 9-7 – Three simulation models used in the materials design problem

9.4.1 Particle Level Shock Simulation Model

The particle level shock simulation model is a microscale finite element simulation that provides spatial resolution of the coupled thermal, mechanical, and chemical responses at the particle level during shock compaction. The model is developed by Austin in his MS thesis (Austin 2005). The details of the model are adapted from his MS thesis. This model is used to incorporate the effect of changing size of constituent particles (aluminum and iron-oxide), different arrangements of the particles in space, subjected to different loading conditions the overall properties of the material at macroscale. The model is used to predict the average temperature at hot spots, equation of state properties of the material, the size of hot spots, and the number of reaction sites. The inputs and outputs of the model are shown in Figure 9-8. Some details of the model are discussed next.

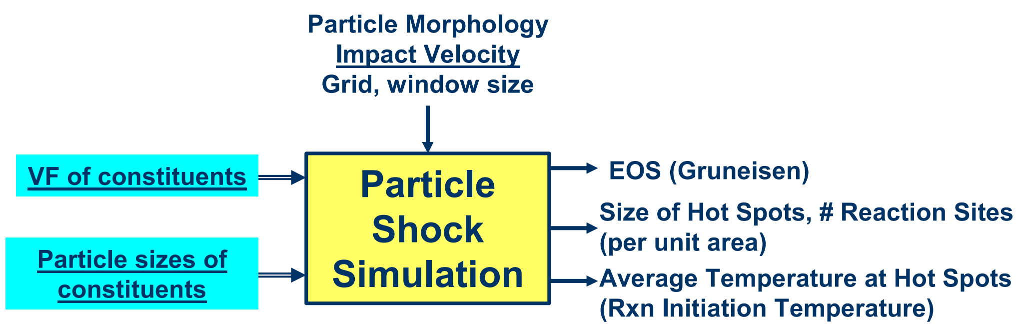


Figure 9-8 - Inputs and outputs for particle level shock simulation model

The first step in the shock simulation is the generation of synthetic microstructures, based on experimental data. Information obtained from microscopy of RPMMs is used to generate size distribution of the particles and voids, which is a lognormal distribution. Experimental data is also used to generate nearest-neighborhood distribution of particles. This information about size and particle distributions is used to randomly generate discrete sets of micron-scale particles (aluminum particles, iron oxide agglomerates, and voids). The particle size is controlled based on the mean and variance values of particle sizes observed from the microscopic images and the generation of number of particles is controlled by the prescribed volume fractions of the statistical volume element (SVE) under consideration. The distribution of particles in the SVE is controlled by the nearest neighborhood distributions. Since the 3-D structure is modeled as a 2-D structure with circular particles, small amount of overlap is permitted. The remaining part of the SVE is filled with epoxy.

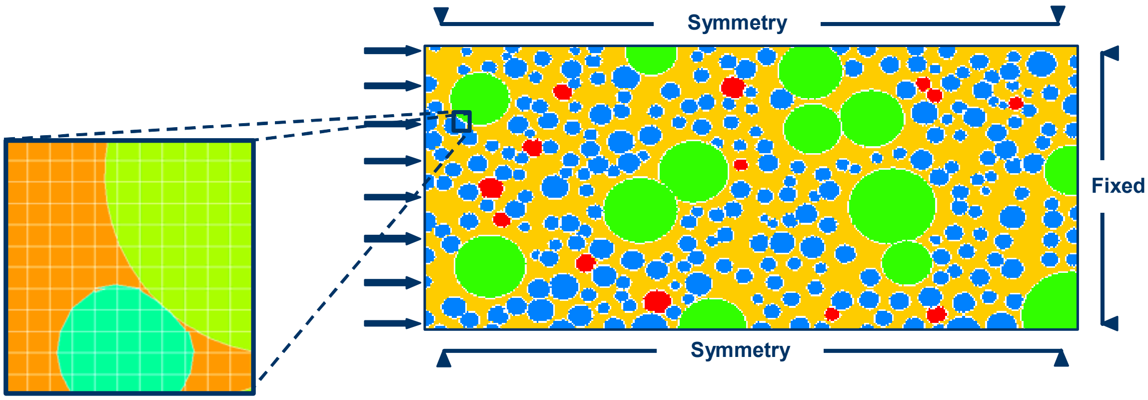


Figure 9-9 - Boundary conditions of the discrete particle shock simulation (Austin 2005)

After the particle structure is generated, the next step is to perform numerical simulation using finite element techniques. In this model, shock waves are propagated through the reactive particle systems to understand the thermo-mechanical conditions that lead to reaction initiation. The simulation is performed using an Eulerian hydrocode Raven (Benson 1995). The boundary conditions on the SVE are shown in Figure 9-9. The shock propagation phenomenon is idealized as a 1-D shock wave. A compressive shock wave is propagated through the mixture by applying a Lagrangian velocity boundary condition to the left surface of the SVE. The velocity of particle is represented as Up. Symmetry planes serve as Lagrangian boundary conditions for the top and bottom surface of the model. A fixed Lagrangian boundary condition is imposed on the right hand side surface. The simulation is carried out until the shock wave propagates 95% of the SVE to avoid wave reflections.

The material properties are modeled in terms of the hydrostatic and deviatoric components of stress-strain response. Mie-Gruneisen Equation of State (EOS) (Kinslow

1970; Asay and Shahinpoor 1993) is used to model the hydrostatic response of aluminum and epoxy phases; and Murnagham EOS (Murnaghan 1937) is used to model the hydrostatic model for the iron oxide phase. A physically based constitutive model proposed by Klepaczko (Klepaczko, Sasaki et al. 1993) is used to model the deviatoric stress-strain response of the aluminum phase. The Hansen-Boyce model (Hasan and Boyce 1995) is used as the strength model for the epoxy phase. The details of the material models are not included in this dissertation because the focus here is not on modeling, but on use of this model for design. For details of the model, refer (Austin 2005).

The performance of the reactive particle systems is evaluated based on *a)* the number of sites where reaction initiates during shock propagation, *b)* the average temperature at the hot spots, and *c)* the hydrostatic behavior of the overall mixture. The reaction initiation is possible where the reactants are in intimate contact. The initiation of reaction is characterized by unbounded growth of hot spots that develop at reactant interfaces due to the heat liberated by exothermic chemical reactions. The reaction initiation predicted using the shock simulation model is at the microscale level, which is different from reaction propagation, which is at a macroscopic level. Reaction propagation is not predicted using the particle shock simulation model. The prediction of reaction initiation conditions is based on the Merzhanov criterion (Merzhanov 1966), according to which, the thermal explosion of hot spots occur when the heat generated by chemical reaction is greater than the heat conduction to the surroundings. The factors affecting the reaction initiation criterion include the temperature at the hot spots, the temperature of the hot spot surroundings, and the size of the hot spots. The maximum number of reaction initiation sites is calculated along different time steps representing the shock propagation. One of the outputs of the particle shock simulation is the temperature at various points in the domain and the size of hot spots. The temperature at various hot spots along with the area of hot spots is used to calculate the area weighted average of the hot spot temperature as follows:

$$T_{ignit} = \frac{\sum_{i=1}^{n} A_i \cdot T_i}{\sum_{i=1}^{n} A_i}$$

where, n is the number of hot spots, T is temperature of a hot spot, and A is size of a hot spot. It is assumed that the main criterion for determining chemical reaction initiation is temperature. Note that this is an approximation of the reaction initiation criterion. This weighted average of temperature is calculated at the time step when the first reaction starts anywhere in the domain. In Figure 9-10, hot spots where reaction is initiated are illustrated in a temperature distribution profile at the time when the first reaction starts. For example, the temperature profile shown in Figure 9-10 is captured at 0.66 nano-seconds when the first reaction initiation hot spots (i.e., three spots) appear. The critical temperature at which chemical reaction will be initiated is the average of the hot spot temperatures with weighting by the spot sizes; this weighted average temperature is the input parameter in the NTMM as the reaction initiation condition. The average hot spot temperature is used in the non-equilibrium thermodynamic mixture model discussed in Section 9.4.2.

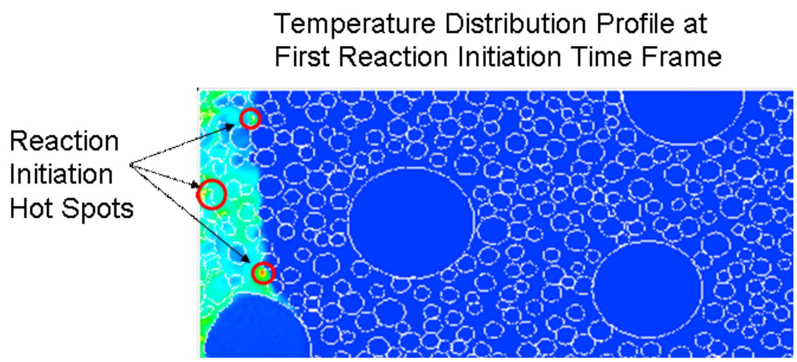


Figure 9-10 - Local hot spots at a first reaction initiation in the reactive particle mixture

The particle shock simulation model is also used to determine the effect of changing material properties and morphology on the hydrostatic behavior of the overall material in terms of the Gruneisen Equation of State (EOS). An equation of state describes the relationship between the pressure, mass density, and internal energy of a material, e.g., $P = P(\rho, e)$ (Austin 2005). Since the operating conditions of the material lie in high pressure range, simple linear elastic relations are unsuitable. In such conditions, Gruneisen EOS is widely used. Due to the complexity, the details of Gruneisen EOS are not presented here. Interested readers are pointed to Chapter 5 in (Austin 2005). The only point of relevance in the discussion of this chapter is that the parameters for Gruneisen EOS can be calculated by performing a linear regression on shock wave speed-particle speed data. The slope (S) and intercept (C) of this regression line are used in the Gruneisen EOS model for the material. These two parameters are useful in the material model in projectile level simulation. As a summary, the inputs and outputs of the simulation code for the sake of design are shown in Figure 9-8. The simulation code is executed at various points in the design space using a design of experiment. The variation in response due to changing material morphology is captured by generating different particle distributions and executing the model multiple times at a given point in the design space (which is specified by the values of design variables). The response surfaces for Gruneien EOS for the mixture and the average hot-spot temperature are provided in the Table 9-4.

Table 9-4 – Response surface equations for Gruneisen EOS parameters, and the average hot spot temperatures

*S = 1.45000 + Ra_Al * (-0.07417) + Ra_Fe2 * (-0.02417) + Ra_Void * (-0.11583) + Vf_void * 0.16250 + Ra_Al*Ra_Al * (-0.03208) + Ra_Fe2*Ra_Fe2 * (-0.10708) + Ra_Void*Ra_Void * 0.05792 + Vf_void*Vf_void * 0.06792 + Ra_Al*Ra_Fe2 * 0.00750 + Ra_Al*Ra_Void * (-0.17250) + Ra_Al*Vf_void * (-0.01750) + Ra_Fe2*Ra_Void * (-0.25750) + Ra_Fe2*Vf_void * 0.07750 + Ra_Void*Vf_void * (-0.11250)* *C = 2.59000 + Ra_Al*0.19667 + Ra_Fe2 * 0.09333 + Ra_Void * 0.17500 + Vf_void * (-0.26667) + Ra_Al*Ra_Al * 0.00583 + Ra_Fe2*Ra_Fe2 * 0.00083 + Ra_Void*Ra_Void * (-0.09917) + Vf_void*Vf_void * (-0.06917) + Ra_Al*Ra_Fe2 * 0.00500 + Ra_Al*Ra_Void * 0.16000 + Ra_Al*Vf_void * 0.02500 + Ra_Fe2*Ra_Void * 0.25500 + Ra_Fe2*Vf_void * -0.08000 + Ra_Void*Vf_void * 0.15500* *Where,* *S is the slope in the Gruneisen EOS* *C is the intercept in Gruneisen EOS* *Ra_Al is the radius of aluminum particles normalized to (-1 and 1) between 0.0005 and 0.0015mm* *Ra_Fe2 is the radius of iron oxide particles normalized to (-1 and 1) between 0.0002 and 0.0010mm* *Ra_Void is the radius of voids normalized to (-1 and 1) between 0.0002 and 0.0010mm* *Vf_void the volume fraction of voids and ranges between 0.02 and 0.10*
*y = (0.057632) + (1.0566) * Ra_Al + (-41.796) * Ra_Fe2 + (-0.28438) * Vf_void+ (-33.785) * Ra_Void + (986.21) * Ra_Al * Ra_Al + (29929) * Ra_Fe2*Ra_Fe2 + (1.9563) * Vf_void*Vf_void + (13711) * Ra_Void*Ra_Void + (2270) * Ra_Al*Ra_Fe2 + (-74.761) * Ra_Al*Vf_void + (1351.5) * Ra_Al*Ra_Void + (-55.384) * Ra_Fe2*Vf_void + (10091) * Ra_Fe2*Ra_Void + (195.5) * Vf_void*Ra_Void* *HotSoptTemp_avg = 1000* (y^(-1/3) - 2)* *Where,* *HotSoptTemp_avg is the average hot sopt temperature generated after the shock passes through the material* *Ra_Al is the radius of aluminum particles normalized to (-1 and 1) between 0.0005 and 0.0015mm* *Ra_Fe2 is the radius of iron oxide particles normalized to (-1 and 1) between 0.0002 and 0.0010mm* *Ra_Void is the radius of voids normalized to (-1 and 1) between 0.0002 and 0.0010mm* *Vf_void the volume fraction of voids and ranges between 0.02 and 0.10*

9.4.2 Non-equilibrium Thermodynamics Mixture Model

In the non-equilibrium thermodynamics mixture model, shock-induced chemical reactions in aluminum and iron-oxide mixtures are modeled in the framework of non-equilibrium thermodynamics and continuum mechanics, in which both thermo-chemical and mechano-chemical processes are accommodated (Lu, Narayanan et al. 2003). The discussion in this section is adapted from (Choi 2005). The constitutive model and the conservation equation are formulated by introducing a combination of internal state variables and extended irreversible state variables. The internal state variables are mass fractions of reactants and products, and void contents. The extended irreversible state variables include chemical reaction rate, heat flux, and pore collapse flux. The irreversibility of these processes are implied in the nonnegative entropy production rate (i.e., the second law of thermodynamics) and their contribution to the dissipation. Relaxation times during to the duration of the chemical initiation and sustained reactions are in the range of 100-200 nano-seconds. A uniformly blended mixture theory is used to describe the porous mixture. The chemical reaction of the constituents is described as –

$$2Al + Fe_2O_3 \rightarrow 2Fe + Al_2O_3$$

Conservation equations, constitutive models, and chemical reaction equations are described in detail in (Lu, Narayanan et al. 2003). The simulation model is implemented in MATLAB®. The example is shown in Figure 9-11.

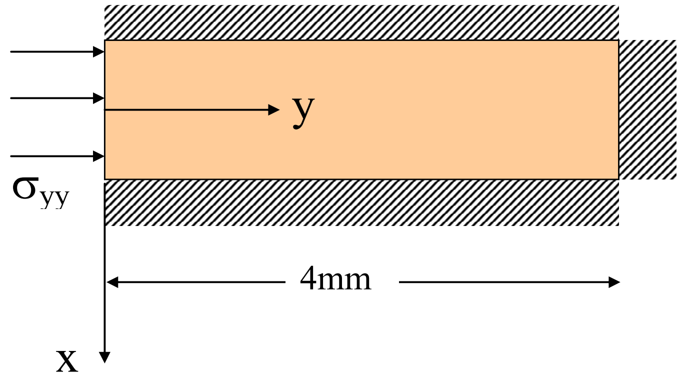


Figure 9-11 One dimensional shock simulation of Non-equilibrium Thermodynamics Mixture

Top, bottom, and right boundary condition is fixed and initial loading (σ_{yy}) is applied on the left boundary. In Table 9-5, the input and output parameters of the non-equilibrium thermodynamic mixture model are listed and those variables implemented in the MATLAB® code are listed.

Table 9-5 Input and output parameters in NTMM analysis

Input Parameters	Output Parameters
	▪ Mass fraction of Al: c_al
▪ Volume fraction of Al: v_al0	▪ Mass fraction of Fe_2O_3: c_fe2o3
▪ Volume fraction of Fe_2O_3: v_fe2o30	▪ Mass fraction of Fe: c_fe
▪ Volume fraction of Fe: v_fe0	▪ Mass fraction of Al_2O_3: c_al2o3
▪ Volume fraction of Al_2O_3: v_al2o30	▪ Pressure: P
▪ Porosity: alpha0	▪ Temperature: theta
▪ Applied loading: σ_{yy}	▪ Porosity: alpha
▪ Initial Temperature: theta0	▪ Density: rho
▪ Reaction initiation criterion: thetac	▪ Stress: sigmax, sigmay
	▪ Velocity :vy
	▪ Heat flux :q

The results of a NTMM execution are illustrated in Figure 9-12. The initial conditions are as follows.

- Loading pressure (σ_{yy}) = 15 (GPa)
- Initial porosity (alpha0) = 1.5

- Initial volume fraction of Al (v_al0) = 0.2545
- Initial volume fraction Fe_2O_3 (v_fe2o30) = 0.7455
- Initial volume fraction of Al_2O_3 (v_al2o30) = 0
- Initial volume fraction of Fe (v_fe0) = 0
- Initial temperature (theta0) = 300 (K)
- Reaction initiation criteria (thetac) = 700 (K)

The results shown in Figure 9-12 are the distributions of pressure, temperature, and mass fraction of Fe at the time frame of 300 nano-seconds after the initial loading.

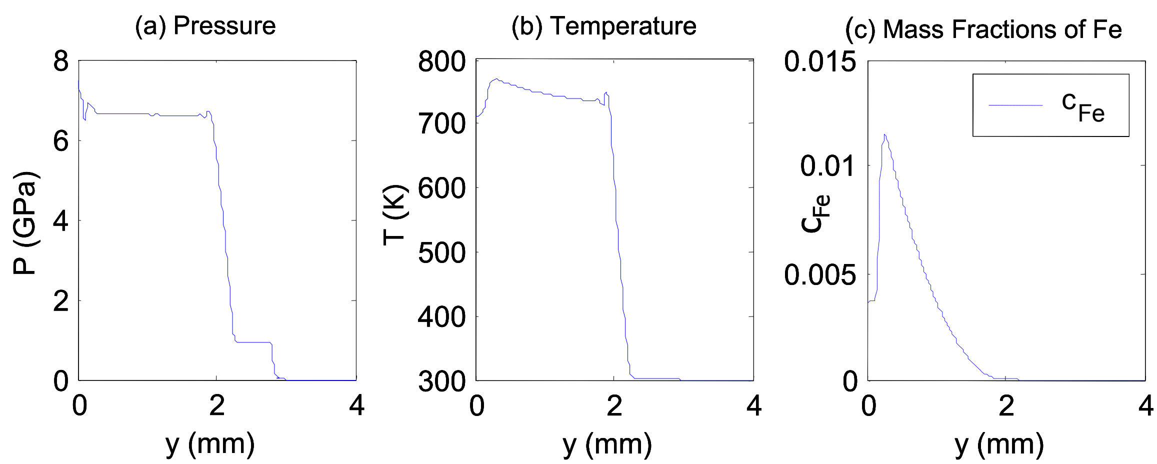


Figure 9-12 An example of NTMM execution

The output that we need to focus on in this analysis is the amount of chemical reaction in the material system. In order to assess the amount of chemical reaction, the mass fraction of Fe is the parameter to be captured since it is the product of the chemical reaction. In this study, we calculate the sum of the predicted mass fraction of Fe at all nodes in the finite difference meshes in the non-equilibrium thermodynamic mixture model at 300 nano-seconds after the initial loading. This parameter is called as the accumulated mass fraction of Fe (*acFe*) in this dissertation.

In summary, the simulation model is a non-equilibrium thermodynamic model incorporating shock-induced chemical reactions. In this model, void collapse flux, chemical reaction flux, heat flux and associated relaxation times in the constitutive models are included, which explains the delayed initiation and sustained chemical reaction. However, reaction initiation conditions in the model are assumed and these reaction initiation criteria need to be obtained from the lower scale model, microscale discrete particle mixture model, to predict more accurate simulation results. The inputs and outputs for the simulation model are shown in Figure 9-13. The discussion of flow of information between the models is presented in Section 9.4.4. Note that the details of the simulation models are beyond the scope of this dissertation. The non-equilibrium simulation code is executed for different values of the volume fraction of constituents and average temperature of hot spots. A response surface of accumulated iron as a function of the inputs for the models is provided in the Table 9-6.

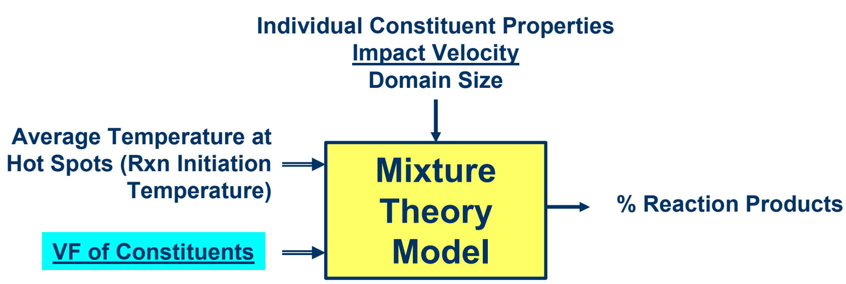


Figure 9-13 - Inputs and outputs for mixture theory model

Table 9-6 – Response surface of accumulated iron as a function of mixture theory model inputs

*accFe = 1.731 - 266.985 * Vf_void + .066639 * T_ignit - 237.500 * Vf_void * Vf_void - 46.522e-6 * T_ignit * T_ignit + 0.283833 * T_ignit * Vf_void;* *Where,* *accFe is the content of accumulated iron* *T is the temperature (in deg K) at which the reaction initiates* *Vf_void the volume fraction of voids and ranges between 0.02 and 0.10*

9.4.3 Projectile Level Simulation

The first two models presented in Sections 9.4.1 and 9.4.2 are material level models that predict the performance of material. The models facilitate understanding the effect of changing composition and morphology of material on the overall *material* properties. In contrast to these material level models, the projectile level model is a system level model that allows designers to vary the system level parameters such as system dimensions, projectile velocity, projectile nose shape, penetration angle, etc. The objective of the projectile level model is to simulate the effect of these system parameters on the overall system performance. The overall performance constitutes the reaction initiation, propagation and the strength of the system. The reaction initiation behavior of the material is modeled in the particle shock simulation model (see section 9.4.1) and the reaction propagation behavior is discussed in the non-equilibrium mixture theory model (see section 9.4.2). In addition to the reaction behavior, another important requirement for the projectile is the ability to *withstand loads that result from high striking velocity impact of selected solids* (see the requirements statement in Section 9.2). The ability to withstand loads is a function of the strength of the overall system, which is dependent on the dimensions of the projectile, the amount of MESMs used, the angle of attack, the impact velocity, and the material properties (Ballew 2004). The loads generated during the impact are also dependent on the target material. Hence, different projectiles should ideally be designed for different targets. Since the projectile level simulation model for the MURI project is under development during the time this dissertation is written, a simplified model, for the purpose of this dissertation, is developed in LS-Dyna® to incorporate the effect of system level parameters on the strength of the overall system. The simplified model developed for this dissertation is discussed in this section.

The simplified model developed for this dissertation is used to simulate a Taylor Impact test (Taylor 1946; Taylor 1948). In the Taylor Impact test, a cylindrical projectile is provided an initial velocity and impacted against a rigid wall. Due to the impact, the impacting end deforms into a 'mushroom' shape. The amount of deformation and the size of the 'mushroom' are dependent on the material properties, the initial velocity and the projectile dimensions. Hence, the test results serve as an indicator of the strength of the projectile. In the simulation model developed for this dissertation, the cylindrical projectile consists of an outer hollow steel shell filled with the MESM material which is designed. The outer diameter of the steel cylinder is fixed to 50mm. The length of the projectile is fixed at 100mm. The inner diameter of the steel shell is a design variable and is assumed to vary between [10 46]mm. The impact velocity is fixed to 1000m/sec. A section of the projectile is shown in Figure 9-14. This projectile is impacted against a rigid wall. The impact is simulated using an explicit Finite Element code in LS-Dyna. The deformation of the projectile is measured after a predefined fixed time (t = 5 μsec in this case). A sample final shape of the projectile is shown in Figure 9-15. The maximum radius of the deformed shape is measured. This maximum radius of the deformed shape is an indicator of the strength of the projectile. The inputs and outputs for the model are shown in Figure 9-16.

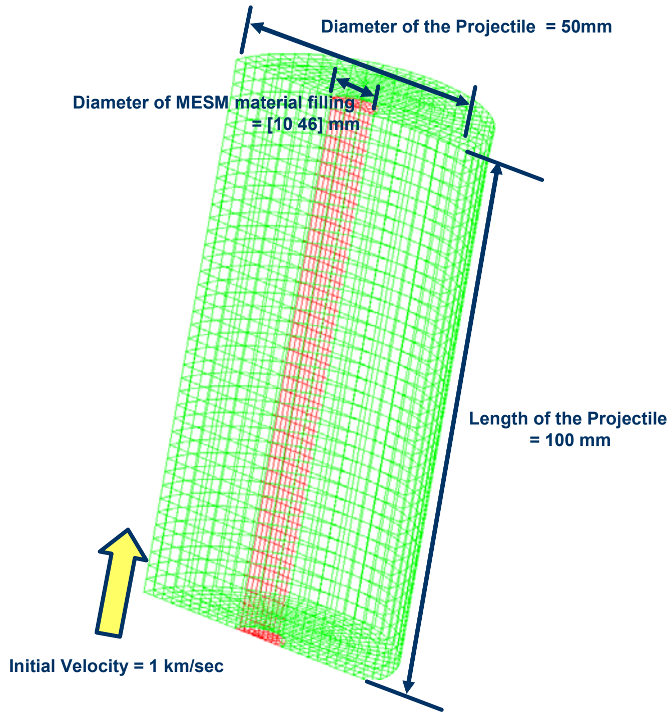


Figure 9-14 - Section of the projectile used for Taylor Impact test in LS-Dyna

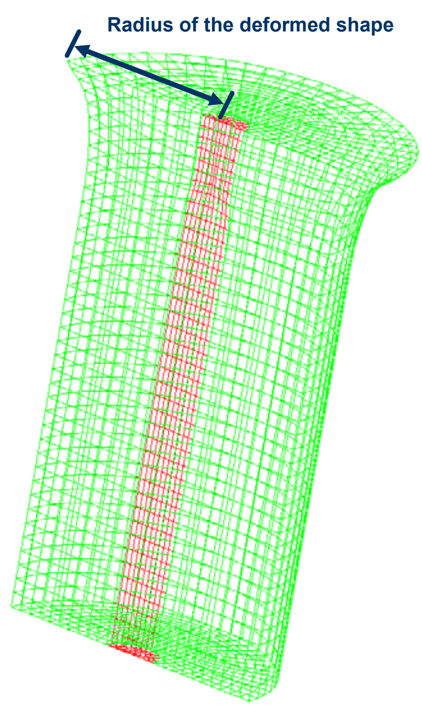


Figure 9-15 – A sample shape of the projectile after the impact

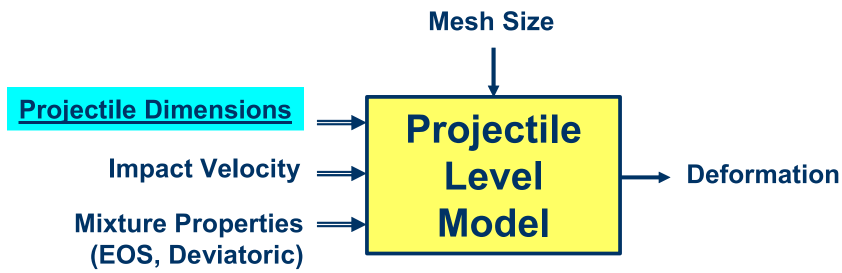


Figure 9-16 - Inputs and outputs for projectile level model

The projectile level model is developed based on the elastic-plastic behavior of iron. The behavior of the MESM material is modeled as a combination of the hydrostatic and deviatoric behavior. The hydrostatic behavior is modeled using a Gruneisen Equation of State (see the description of Gruneisen EOS in Section 9.4.1). The devatoric behavior of the material is modeled using the experimental data reported by Patel (Patel 2004). For the Taylor Impact model developed in this dissertation, the deviatoric behavior of the model is assumed to be constant for different values of material parameters such as size of constituent particles, volume fraction of voids, etc. The variation in material properties is incorporated in the model by considering the changes in material's hydrostatic behavior – i.e., the changes in parameters of the Equation of State. Note that the projectile level model used in this dissertation is extremely simplified. The model does not account for effects such as shape of the nose, penetration of the target, erosion, friction, heat generation due to reaction, etc. The shape of the projectile is also simplified. However, all these simplifications do not affect the manner in which the method is used. The model can be replaced with a comprehensive model without changing the steps followed in the method. The simulation model is executed for various combinations of the projectile dimensions and Gruneisen EOS parameters for the material to develop a response surface for the deformation as a function of the inputs. The response surface parameters are listed in the Table 9-7.

Table 9-7 – Response surface equations for deformation as a function of projectile level simulation inputs

*MaxDeformation = 8.26162 + InnerRadius *(0.30134) + Slope *(0.05052) + Intercept *(0.08575) + InnerRadius*InnerRadius *(-0.34089) + Slope*Slope *(0.00479) + Intercept*Intercept *(-0.00162) + InnerRadius*Slope*(0.09282) + InnerRadius*Intercept*(0.13622) + Slope*Intercept*(0.00696);*
Where, *MaxDeformation is the maximum deformation measured as a result of the impact*
InnerRadius is the radius of the MESM filling in the outer steel shell. This variable is normalized to (-1 and 1) between 5mm and 23 mm *Slope is the slope in Gruneisen EOS and is normalized to (-1 and 1) between 1 and 2* *Intercept is the intercept in Gruneisen EOS and is normalized to (-1 and 1) between 2 and 3*

9.4.4 Linkage Between the Three Material Simulation Models

The three simulation models to be used in this dissertation are discussed in Sections 9.4.1, 9.4.2, and 9.4.3. In this section, we discuss the information flow between the three models. The information flow between the models is shown in Figure 9-17. The first model discussed is a particle shock simulation model. The inputs of the simulation model include volume fraction of various constituents (Aluminum, Iron Oxide, Epoxy, and Voids) and the size distribution of these particles. The outputs of this model include the number of reaction sites, the average size of hotspots, average temperature at reaction initiation, and the parameters for Gruneisen EOS.

The average temperature of reaction initiation is used as a reaction initiation criterion for the non-equilibrium mixture theory model. The average hot spot temperature, in association with the volume fraction of constituents is used to predict the amount of accumulated reaction products, which is an indicator of the extent of reaction propagation. The average hot spot temperature is used to account for the changing material parameters and their morphology in the reaction propagation behavior. The Gruneisen EOS, which is an output from the particle shock simulation, is used as an input for the projectile level simulation for accounting for the changing material properties in the system level simulation. The output of the projectile level simulation is the deformation achieved in the Taylor Impact test. Hence, the three models combined together can be used to predict the strength properties (through deformation from the projectile level model) and the reaction properties (through the accumulated reaction products from the mixture theory model).

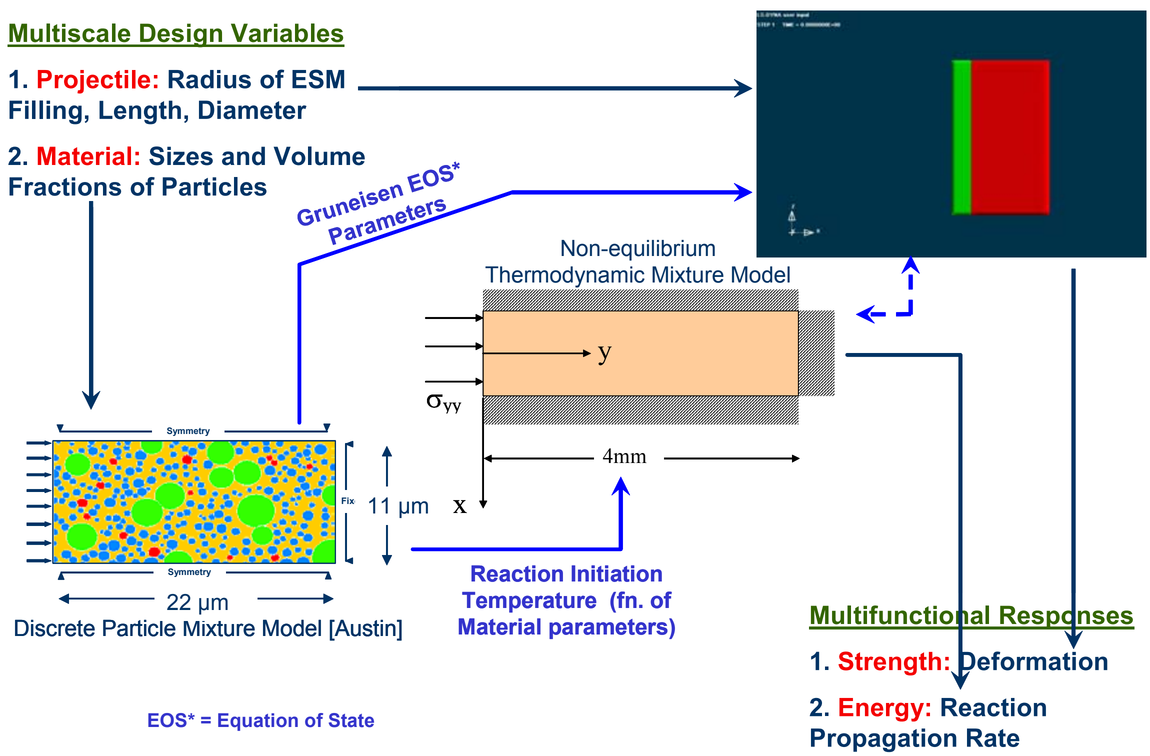


Figure 9-17 - Flow of information between simulation models

9.5 Designing the Material Using Proposed Design Method

In this section, we apply the design method presented in Section 3.5 to the materials design problem. The steps of the design method are presented again for readers' convenience in Figure 9-18. Step 1 of the design process relates to mapping the requirements with simulation models is discussed in detail in 9.5.1. Step 2 in the design method relates to modeling the design processes using interaction patterns, and is discussed in Section 9.5.2. Step 3 relates to simplification of design processes and is discussed in Section 9.5.3. Design process execution and verification is discussed in Section 9.5.4. Simulation model refinement is discussed briefly in Section 9.5.5 and in detail in the following Section 9.6.

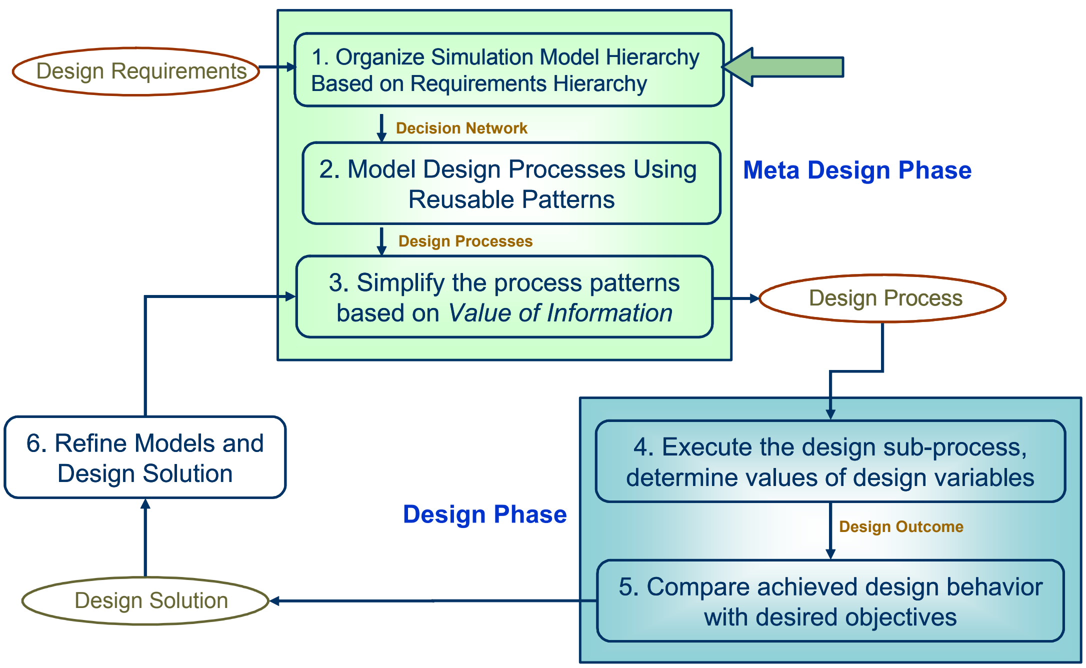


Figure 9-18 – Design method used for integrated design of products, materials, and design processes

9.5.1 Development of Decision Network Based on Available Models

In general, the first step of the method consists of developing a requirements hierarchy and a decision network. The decision network is then mapped to the network of available simulation codes to determine which simulation models are to be used for various decisions. The requirements hierarchy for the multifunctional energetic structural materials is presented in Figure 9-19. The overall requirements are partitioned into individual functional requirements - strength and reaction initiation properties. These functional requirements are then expressed as *performance* metrics such as withstanding impact load, energy release, and reaction time. The performance is then expressed in terms of target values for *properties* such as strain rate, fracture toughness, yield strength, equation of state, temperature-pressure relations for reaction initiation, size of hot spot, etc. This hierarchical structure of requirements, performance and properties is termed as the *requirements hierarchy* (Step 1.1). The target values of properties are achieved by appropriate values of design variables that define the *structure* of the material. These design variables include volume fractions of constituents, particle size, material topology, overall dimensions of the system, etc. In Step 1.2, a relationship between the requirements hierarchy and the structure is developed (see Figure 9-19).

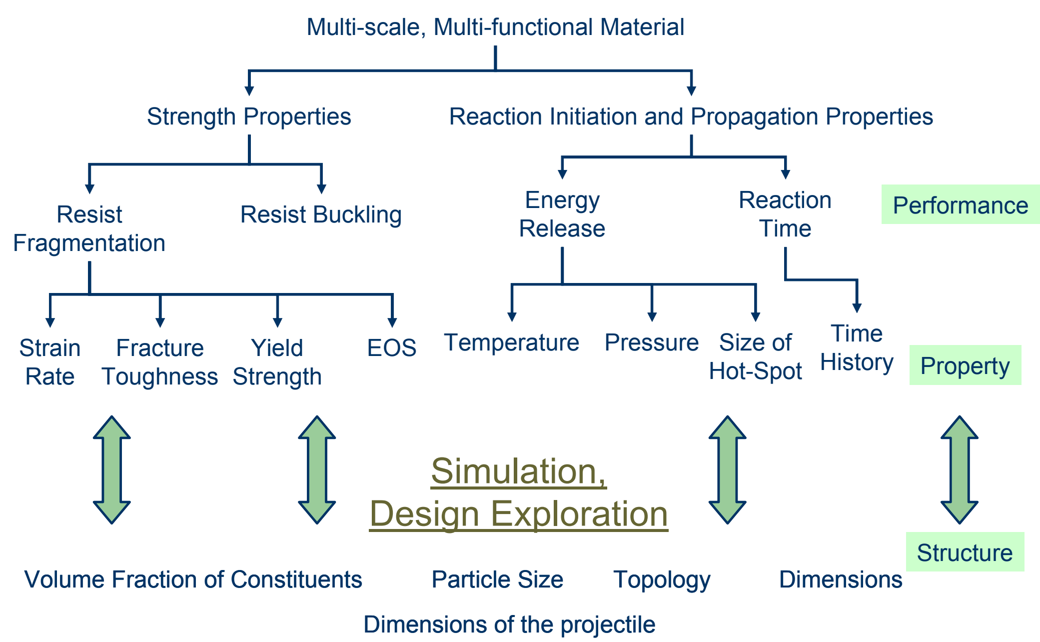


Figure 9-19 - Requirements hierarchy and relationship with structure for materials design problem

Step 1.3 involves laying out the decision network and identifying the information required for each decision. An example decision network for the material design problem is shown in Figure 9-20. The first decision in the decision network shown in the figure is selection of material constituents. Using the information about selected constituents, the properties of individual material constituents can be evaluated. This information can then be used to decide on the relative volume fractions of each material constituent and their spatial distribution. The material mixture can then be simulated to obtain the resulting structural and energetic properties of the mixture. The mixture properties can be used to decide on the dimensions of the overall system such as a projectile and finally, the designers can decide on other system level design variables such impact velocity, angle of attack, etc. Although this is a starting decision network, it can be refined later in the design process. This decision network consists of five decisions. The first three decisions are associated with the material and the last two are associated with the projectile. For the specific problem discussed in Section 9.3, we assume that the material constituents are fixed (we only consider an aluminum, iron-oxide mixture). The impact velocity is fixed to 1000m/sec and the angle of attack is fixed to 90 degrees (i.e., the projectile impacts perpendicular to the target). The parameters for material distribution such as nearest neighborhood distribution are also fixed. Hence, in this specific problem, we have two decisions – material decision (volume fractions and size of constituents), and the projectile decision (projectile dimensions). The problem is simplified for the purpose of this dissertation because the required aspects of the validation of the method, such as decision and scale decoupling, can be shown with these two decisions.

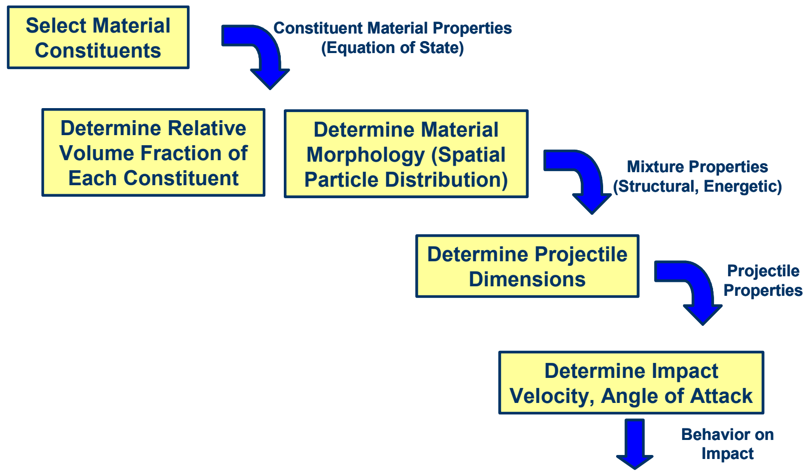


Figure 9-20 – A possible sequence of decisions for materials design

It is assumed that during the start of design process, a number of simulation models are available at different scales of length and time. Some of the examples of models available for designing materials are discussed in Section 9.2. The input/output information for these models is used to determine the flow of information between these models. These models are then organized hierarchically based on their length scales to form a *simulation model hierarchy*. For example, in the materials design example, the simulation models are organized into three groups related to – Hugoniot data generation and validation, reaction initiation prediction, and projective level / Reactive Powder Metal Mixture (RPMM) couplings (see Figure 9-21b). The Hugoniot data is generated from *a)* first principles simulation of lattices and *b)* shock simulation of discrete reactive powder metal mixtures. The Hugoniot data is validated against high strain rate experiments. The reaction initiation criteria are evaluated from ab-initio calculations and are used in interface modeling at MD level and shock simulations of discrete reactive metal powder mixture. The material properties information is then fed into projectile level simulation using response surface modeling techniques.

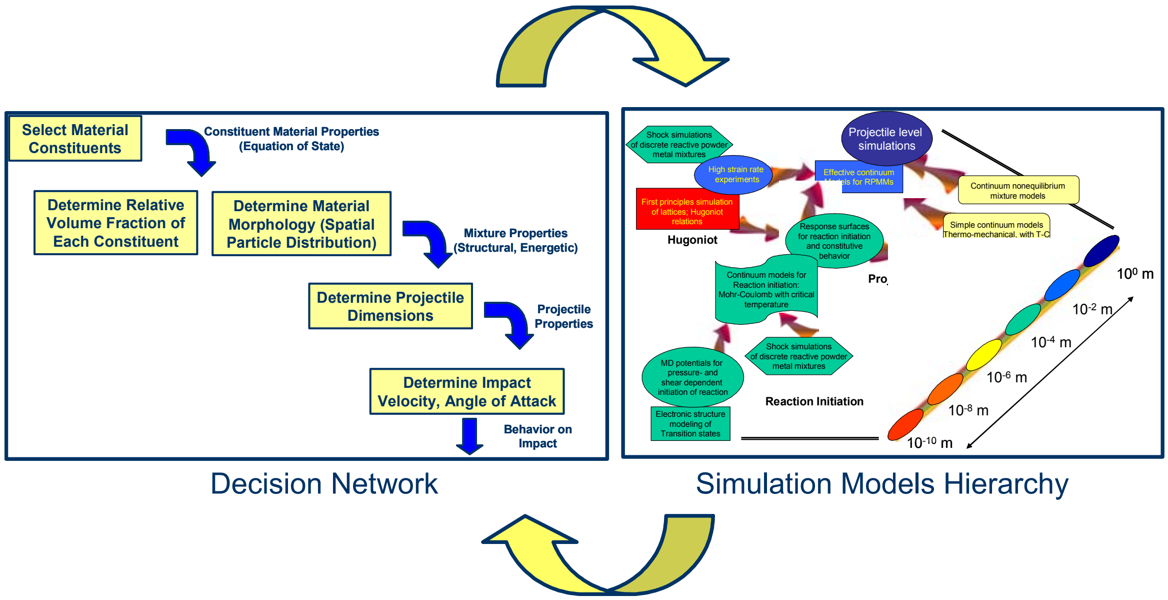


Figure 9-21 – Mapping the requirements hierarchy with simulation model hierarchy

9.5.2 Model Design Processes Using Reusable Patterns

Nine interaction patterns for modeling simulation based design processes are shown in Section 3.5.2. For illustration purposes, examples of these interaction patterns from the materials design domain are presented next. After providing the examples from the general materials design scenario, we present the interaction patterns in terms of the three specific simulation models presented in Section 9.4.

Example of Pattern P1 - Multiscale Models for Validation: Pattern P1 represents the scenario where models at different scales generate similar information. For example, based on the material morphology, an FEM based simulation model (Choi, Austin et al. 2004) is developed for modeling a shock at the mesoscale level in order to simulate the material's Hugoniot data. At a lower scale, a Molecular Dynamics (MD) model is developed to simulate the Hugoniot data based on material morphology at nanoscale level. The results from both these models are compared against each other either for validation purposes or for model calibration (see Figure 9-22) if it is known that one of the models is already valid. This pattern generally occurs in multiscale design where the lower scale model is of high fidelity and hence more accurate, but it is computationally expensive for modeling a larger domain. The higher scale model is computationally efficient, but its parameters need to be calibrated using the lower scale model.

Example of Pattern P5 - Dependent (sequential) responses between multiscale models: In Figure 9-23, we present an example of Pattern P5, where both models are associated with design variables and responses and the flow of information is sequential. The lower level model in this case is particle shock simulation which has volume fractions as inputs and maximum pressure and temperature in the material as outputs. The result of this analysis is Hugoniot data of the overall mixture that is used in developing the constitutive model for the mixture. This information about constitutive model is fed into a system (projectile) level simulation, where design variables are the overall dimensions of the projectile and the response is material behavior on impact.

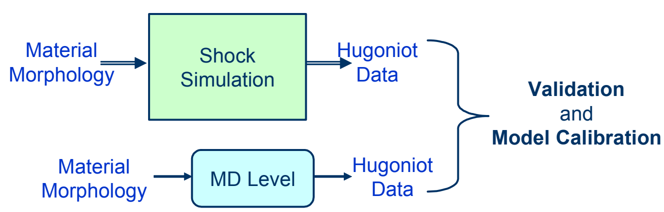


Figure 9-22 – Example of Pattern P1

The standard patterns discussed in this section are used to identify the interaction patterns in the network of models which resulted from mapping the requirements hierarchy and model hierarchy (Step 1 described in the method discussed in Chapter 3). After the design process is developed, the third step in the method is to simplify this process.

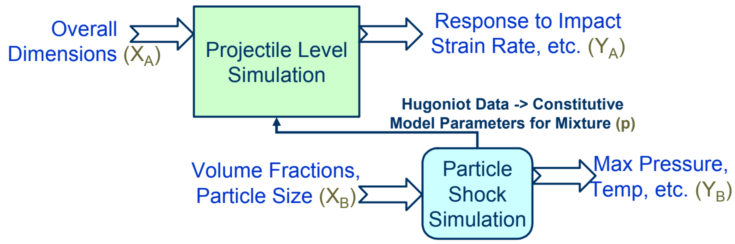


Figure 9-23 – Example of Pattern P5

Simulation model interaction patterns in the design problem under consideration

The examples of interaction patterns discussed so far are patterns with two models in a general multiscale materials design scenario. The problem discussed in this chapter involves three models that exchange information with each other. The flow of information between the simulation models is discussed in Section 9.4.4 and is highlighted in Figure 9-24.

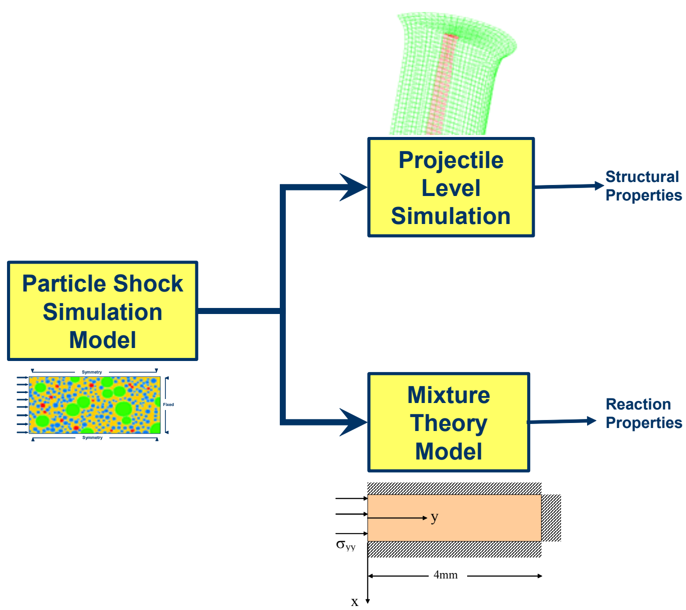


Figure 9-24 – Flow of information between material models

Based on this information flow, the interactions between the *particle shock simulation* and the *projectile level simulation* is sequential, i.e., Pattern P2. Similarly, the interaction between the particle level shock simulation and the non equilibrium mixture theory model is also sequential, i.e., Pattern P2. The patterns are highlighted in Figure 9-25.

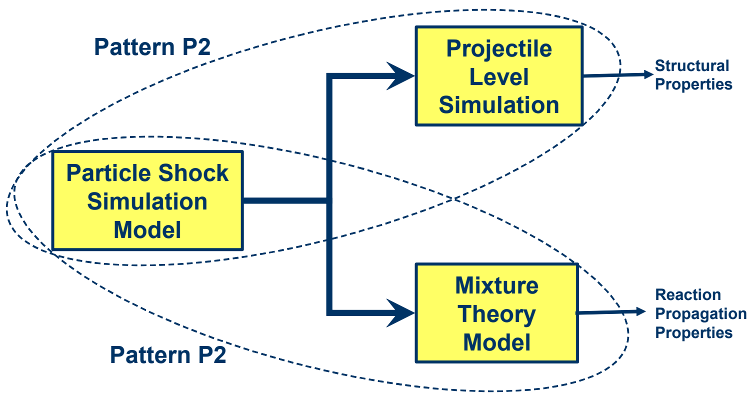


Figure 9-25 – Sequential information flow between the simulation models represented as interaction pattern P2

A sequential interaction patterns can be simplified into an independent interaction pattern if instead of utilizing the output of the first model into the second model, the input of the second model is set to a constant value. The constant value may be the average value of the output from the first model. For example, in the case of interaction between the particle shock simulation (first model) and the projectile level model (second model), the output from the first model is a set of parameters for the equation of state of the material. By varying the inputs of the particle shock simulation, the values of the parameters change. These parameters are fed into the projectile level model to model the effect of varying material properties. If the interaction pattern between the two models is changed to an independent interaction pattern, an average set of values for the equation of state parameters are set as the inputs to projectile level simulation. Hence, the output of the projectile level simulation (deformation) is only a function of the projectile level parameters. Physically, this means that the changes in material properties are ignored during the calculation of deformation. The scenario where the interaction between particle shock simulation and the projectile level simulation is simplified from pattern P2 to Pattern P1 is shown in Figure 9-26. In the figure, the sequential interaction between particle shock simulation and the non-equilibrium mixture theory model is preserved. Following the same logic, four different combinations of interaction patterns between models can be generated. The scenario where both the interaction patterns are simplified from pattern P2 to pattern P1 is shown in Figure 9-27.

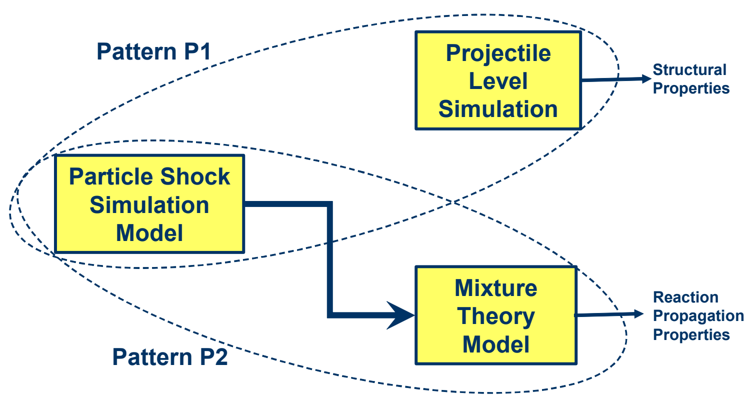


Figure 9-26 – Simplification of interaction between the particle level shock simulation and the projectile level simulation from pattern P2 to P1

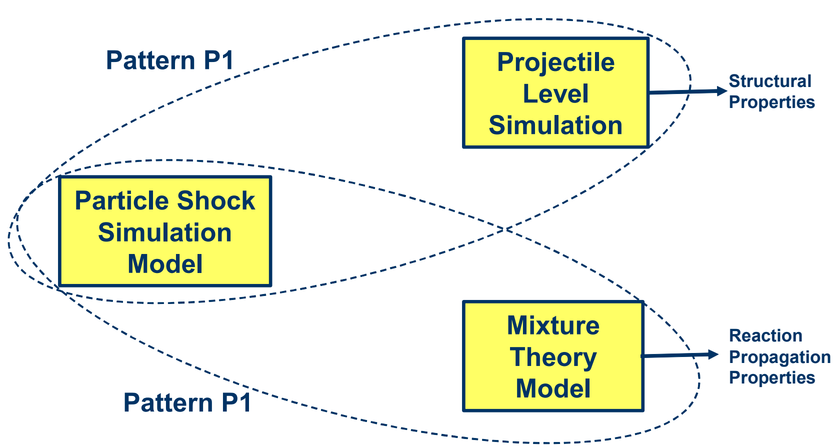


Figure 9-27 – Simplification of both sequential interaction patterns (P2) into independent interaction pattern (P1)

Decision interaction patterns for the design problem under consideration

Analogous to the interaction patterns between simulation models, the decisions about product and material are also associated with interaction patterns (P4 for independent, P5 for sequential and P6 for coupled). Ideally, the decisions about the material and the projectile should be made in a coupled fashion because the design variables associated with both products and materials affect the overall system performance. Various processes (discussed in Section 3.5.4) can be used to make decisions in a coupled fashion. One such process is to make decisions individually and iterate until the solution converges to a single design point. The coupled nature of design decisions and the solution using an iterative process is shown in Figure 9-28. The coupled decisions correspond to interaction pattern P6.

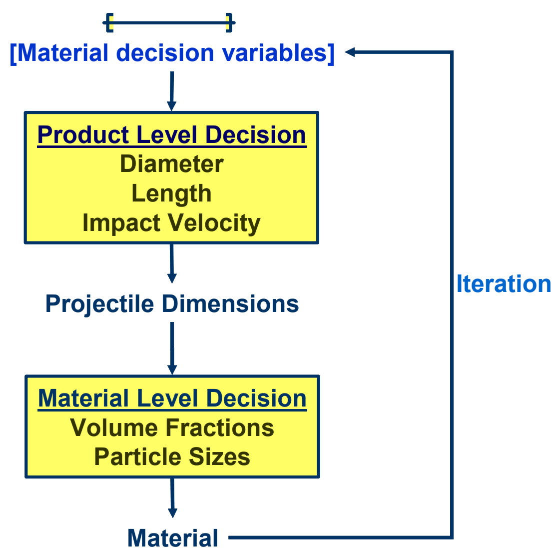


Figure 9-28 - Coupled material and product decision making (Pattern P6)

The decision patterns can also be simplified into a sequential interaction pattern where the decision about projectile (product) level parameters are decided upon first by assuming a set of values for material level design variables. The projectile level parameters are then utilized for making decisions about the material. This interaction pattern assumes that the effect of material parameters on the projectile parameters is insignificant but the effect of projectile level on the material level parameters is significant. This sequential decision making corresponds to the interaction pattern P5 and is shown in Figure 9-29. The sequence of decisions can also be reversed by making the material level decision first and then using the information about material level parameters to decide upon the product level parameters. This sequential flow of information also corresponds to pattern P5. Finally, the two decisions can also be made in an independent fashion as shown in Figure 9-30. Hence, there are four different configurations in which the decisions about product and material can be made (coupled, sequential with material decision first, sequential with product decision first, and independent decision).

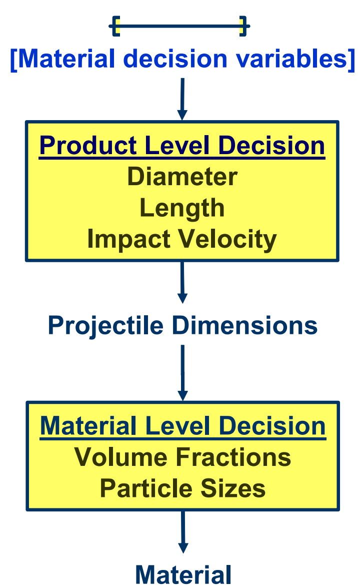


Figure 9-29 – Sequential decision making (Pattern P5)

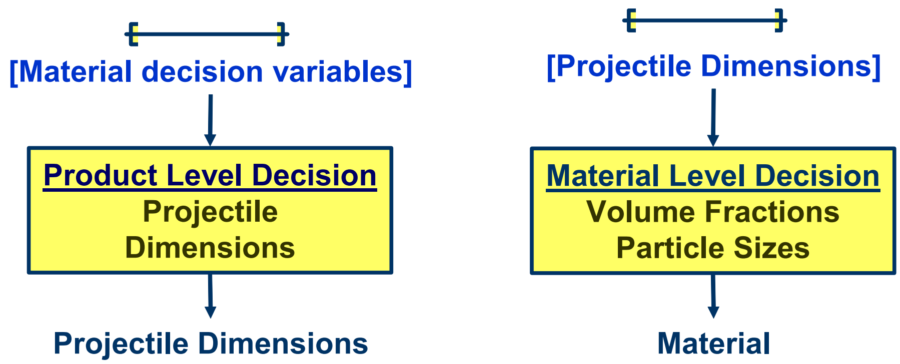


Figure 9-30 – Independent decision making - Pattern P4

DSM representation of interaction patterns between simulation models and decisions for the design problem under consideration: The information flow between the simulation-models and decisions in this problem are relatively simple and it is convenient to represent the information flows as networks. Complex problems involving more simulation models and decisions can be conveniently modeled using DSM matrix representation presented in Section 3.3.2. Using the DSM representation, the interaction patterns between models and decisions can be identified when the infomation flow is more complex (see discussion in Section 3.5.2). However, just for illustration purposes, the interaction patterns between the models can be represented using the DSM matrix form as shown in Figure 9-31. The variables labeled as x_i are used to represent sets of design variables. For example, x_1 corresponds to all the design variables associated with size and volume fractions of constituents. Similarly, the variables y_i correspond to the response variables. The analysis models are represented as A_i and the decisions are represented as D_i. Note that the decisions are coupled with each other because the shaded sub-matrix corresponding to the decisions is not lower-triangular. The analyses are sequential in nature as is evident from the lower triangular nature of shaded matrix corresponding to analyses.

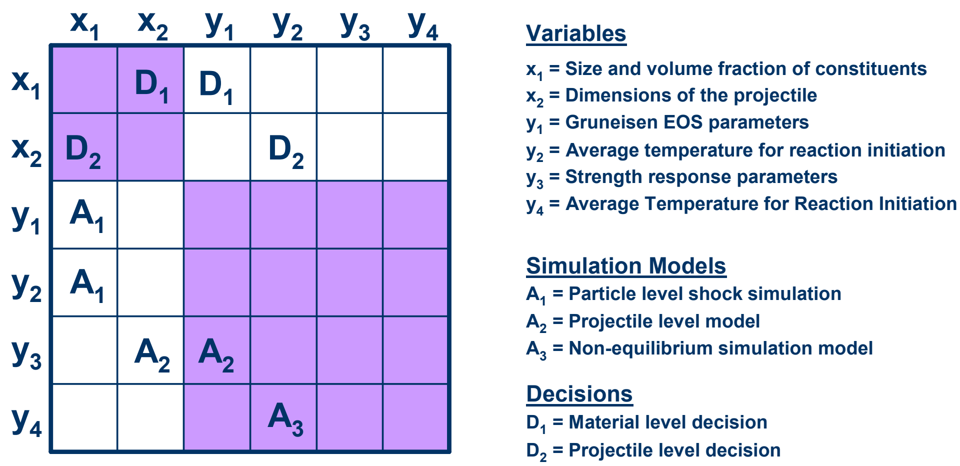


Figure 9-31 – Two decisions and three simulation models in materials design problem represented in a DSM matrix form

9.5.3 Design Process Simplification

Note that in Section 9.5.2, four interaction patterns for simulation models are shown along with four separate interaction patterns for the simulation models. These interaction patterns for simulation models and decisions can be combined together because the simulation models are used to make decisions (see Figure 9-32). At the design process level, the designers need to make two decisions – *a)* which interaction pattern should be used for decisions? and *b)* which interaction pattern should be used for the simulation models? These two process level decisions are also coupled with each other. The choice of simulation model interactions determines which decision interaction pattern is appropriate and vice versa. Since there are four different types of model interaction patterns and four different types of decision interaction patterns, we have a total of 16 alternatives for the decision related to simplification of design process. These 16 design process related alternatives are labeled A through G in a matrix form in Figure 9-33. Each row in the matrix corresponds to a specific model interaction pattern and each column corresponds to a specific decision interaction pattern. Each cell in the matrix corresponds to a unique combination of model and simulation interaction patterns. The correspondence of each cell in the matrix with interaction patterns is shown in Table 9-8.

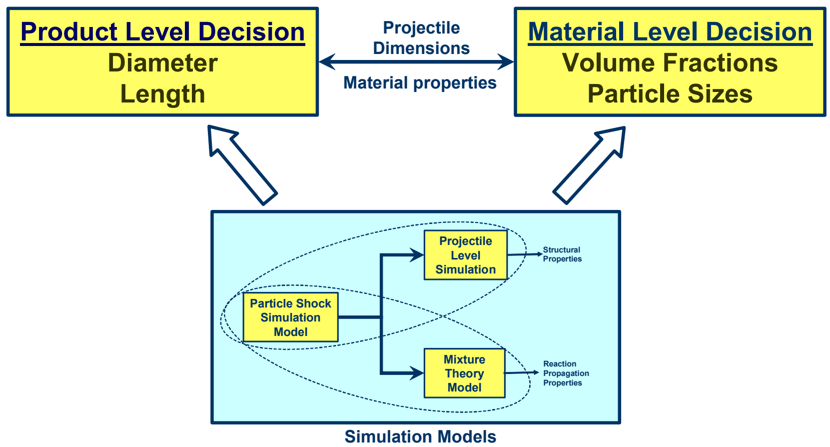


Figure 9-32 - Relationship between the model and decision interaction patterns

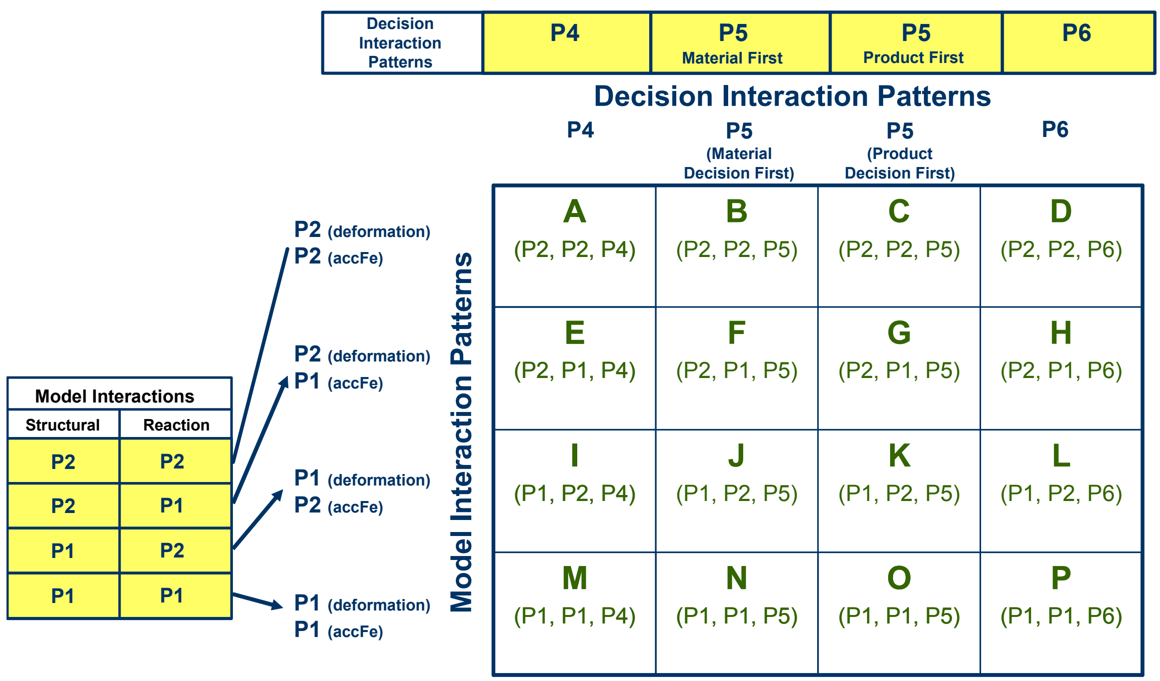


Figure 9-33 – Alternatives (A through P) for design process simplifications

Table 9-8 - Interaction patterns between decisions and simulation models along with associated labels

Scenario	Interaction Patterns		
	Decision	Simulation Model	
	Decision Interaction	Pattern_deformation	Pattern_accFe
A	P4	P2	P2
B	P5 (Material First)	P2	P2
C	P5 (Product First)	P2	P2
D	P6	P2	P2
E	P4	P2	P1
F	P5 (Material First)	P2	P1
G	P5 (Product First)	P2	P1
H	P6	P2	P1
I	P4	P1	P2
J	P5 (Material First)	P1	P2
K	P5 (Product First)	P1	P2
L	P6	P1	P2
M	P4	P1	P1
N	P5 (Material First)	P1	P1
O	P5 (Product First)	P1	P1
P	P6	P1	P1

9.5.4 Design Process Execution

The results from execution of design processes using the different combinations of model and decision interaction patterns are shown in Table 9-9, Figure 9-34, Figure 9-35, and are discussed in this section. In Table 9-9, the outcomes of decisions corresponding to each combination (A through P) are shown in individual rows. The columns in the table list the corresponding overall utility values at the decision point (defined in Table 9-3 and graphically illustrated in Figure 9-5), the values of design variables at the design point, and the corresponding response values. The design variables include four material level variables – *a)* size of aluminum particles (SizeAl), *b)* size of iron oxide particles (SizeFe2O3), *c)* size of voids (SizeVoids), *d)* volume fraction of voids (VF voids), and the product (projectile) level design variable – radius of filling of the MESM material. The response values are average deformation in the projectile at the predefined time (Deformation_avg) and the average mass fraction of accumulated reaction product (AccFe_avg). The table is used to compare the decisions made using each combination of interaction patterns labeled A through P.

Table 9-9 – Utilities, responses, and design variable values for various interaction pattern scenarios

Scenario	Utilities	Responses		Design Variables				
	Overall			Material				Product
	Util_Overall	Deformation_avg	AccFe_avg	SizeAl	SizeFe2O3	SizeVoids	VFVoids	RadFilling
A	0.926	8.195559	12.027799	0.0005	0.0006	0.0007	0.05	23
B	0.926	8.195559	12.027799	0.0005	0.0006	0.0007	0.05	23
C	0.941	8.087618	13.406639	0.0015	0.0007	0.0002	0.1	23
D	0.996	7.989223	12.033785	0.0006	0.0006	0.0006	0.05	9
E	0.723	8.077529	12.03358	0.0005	0.0002	0.001	0.04	23
F	0.723	8.077529	12.03358	0.0005	0.0002	0.001	0.04	23
G	0.723	8.077529	12.03358	0.0005	0.0002	0.001	0.04	23
H	0.748	7.99652	12.03358	0.0011	0.001	0.001	0.04	9
I	0.662	8.223549	12.027799	0.0005	0.0006	0.0007	0.05	23
J	0.662	8.223549	12.027799	0.0005	0.0006	0.0007	0.05	23
K	0.662	8.223549	12.027799	0.0005	0.0006	0.0007	0.05	23
L	0.967	7.990475	12.027799	0.0005	0.0006	0.0007	0.05	9
M	0.412	8.223549	12.03358	0.0005	0.0002	0.0002	0.04	23
N	0.412	8.223549	12.03358	0.0005	0.0002	0.0002	0.04	23
O	0.412	8.223549	12.03358	0.0005	0.0002	0.0002	0.04	23
P	0.717	7.990475	12.03358	0.0005	0.0002	0.0002	0.04	9

In Figure 9-34, the upper bounds of ex-post value (discussed in Section 4.3) for each combination is presented in the same matrix format introduced in Figure 9-33.

Decision Interaction Patterns

Model Interaction Patterns	P4	P5 (Material Decision First)	P5 (Product Decision First)	P6
P2 (deformation) P2 (accFe)	(A) 0.7428	(B) 0.0704	(C) 0.0516	(D) 0.0000
P2 (deformation) P1 (accFe)	(E) 0.9157	(F) 0.2757	(G) 0.2647	(H) 0.2500
P1 (deformation) P2 (accFe)	(I) 0.9509	(J) 0.3342	(K) 0.3303	(L) 0.0296
P1 (deformation) P1 (accFe)	(M) 0.9992	(N) 0.9992	(O) 0.9981	(P) 0.5374

Figure 9-34 – Maximum Ex-Post values (value of additional information) achieved for the different combinations of interaction patterns

Recall that this metric quantifies the impact of additional information on a designer's decision making capability and is measured as the difference between maximum payoff that can be achieved using the information and the minimum payoff achieved at the decision point without the information. This metric is used as a basis for making design process related decision – *which combination of interaction pattern is suitable for making decisions about the product and the material?* The values of overall utility achieved at the decision point are shown in the same matrix form in Figure 9-35.

First, we observe the trend in the maximum ex-post values for making process decisions by focusing on the Figure 9-34. We remind that the higher values of ex-post value indicate that there is a large possibility of improvement in the decision through addition of more information, whereas the values close to zero indicate that there is very little benefit that can be achieved in the quality of decision by adding more information. In the context of this section, addition of information is equivalent to moving from decoupled patterns to coupled patterns. For interaction between models, this refers to going from pattern P1 → P2 → P3, whereas for interaction between decisions, this refers to going from pattern P4 → P5 → P6. Based on the ex-post values for each combination of interaction pattern, we observe that:

1. As the designers increase the level of coupling between decisions (from P4 through P6), the maximum ex-post value decreases, which indicates that the possibility of achieving a benefit by increasing the level of coupling between decisions reduces as we move from independent to sequential, and sequential to coupled decisions. This reduction is due to the reduced uncertainty while decision making, and arises from lack of knowledge about other designers' decisions. The ex-post value for the combination 'D' is zero because there is no uncertainty due to simplification in decision interactions or due to simplification of model interactions. The contribution of uncertainty in the overall utility value due to inherent system variability is also zero in this case

(which is just a characteristic of the system and may not be true in all cases). The ex-post value for the combination 'M' is the maximum (=0.9992) because it results from both decision decoupling and model decoupling. Note that in combination 'M', the simplest patterns are used for models and decisions. The trend indicates that more complex interaction patterns result in decisions that are less uncertain, which is inline with the expected trend. The same trend is observed as the designers improve the interaction patterns between models (from P1 → P2).

2. Although the general trend is according to the expectation, more insight into the problem behavior is obtained by looking at the relative values of the maximum ex-post utilities. The independent decision interaction pattern P4 has the highest value of ex-post and the reduction in value from P4→P5 is significantly higher compared to the reduction from P5→P6 in the case where model pattern P2 is used for deformation calculations. For example, if we fix the model interaction pattern to P2 for deformation and P2 for accumulated iron, *a)* the reduction in ex-post value from P4 → P5 is 0.6724 (0.7428-0.0704 = 0.5496) and b) the reduction in ex-post value from P5 → P6 is 0.0704 (0.0704-0.0000 = 0.0704), for the case where material decision is made first. Same is the case when pattern P2 is used for deformation and pattern P1 is used for calculating accumulated iron. From a design process decision making standpoint, this implies that if pattern P2 is used for deformation calculations, the benefit of moving from an independent interaction pattern P4 to sequential model interaction pattern P5 is much more compared to moving from P5 to a

coupled interaction pattern P6. This difference in the reduction reduces as we simplify the model interaction patterns.

3. If the decision about design process is based only on the value of information, the sequential decision interaction pattern P5 where product decision is made first is only marginally preferred over the material first decision.
4. Consider the scenarios where decisions are made in a sequential manner (combinations B, C, F, G, J, K, N, and O). In these combinations, the effect of interaction pattern for deformation calculation has a significant impact on the ex-post value. This is indicated by the high values of the metric when pattern P1 is used for deformation (0.3342, 0.3303, 0.9992, and 0.9981) as against the low values when pattern P2 is used (0.0704, 0.0516, 0.2757, and 0.2647). Hence, from a process decision making standpoint, if the decisions are made in a sequential manner, the deformation should be calculated using a sequential pattern for deformation.
5. If maximum ex-post value is the only criterion for metalevel decision making, then the designers would choose combinations B,C,D, and L. All these patterns result in an ex-post value of information that is less than 0.100. In other words, the maximum possibility of improvement in the payoff by addition of information is less than 0.100. These seven combinations are highlighted in Figure 9-34.

It is important to note that ex-post value is an important metric to consider while making design process decisions; it is not the only metric. While making design process related decisions, designers should also consider other factors such as design freedom, robustness of process, complexity of the process, cost of executing the process, etc. Due to the scope of this dissertation, these factors are not included. These factors are discussed in the future work section in Chapter 10. One such factor is the payoff achieved at the decision point. The effect of payoff achieved at the design point is mentioned in Section 4.3 in the context of opportunity and achievement ratios. To keep this discussion of the results simple, we look at the achieved utility values only. Conclusions similar to the following discussion can also be derived from achievement and opportunity ratios. The achieved utility values for different combinations of model and decision interaction patterns are shown in Figure 9-35 and discussed in the following.

Model Interaction Patterns \ Decision Interaction Patterns	P4	P5 (Material Decision First)	P5 (Product Decision First)	P6
P2 (deformation) P2 (accFe)	(A) 0.926	(B) 0.926	(C) 0.941	(D) 0.996
P2 (deformation) P1 (accFe)	(E) 0.723	(F) 0.723	(G) 0.723	(H) 0.748
P1 (deformation) P2 (accFe)	(I) 0.662	(J) 0.662	(K) 0.662	(L) 0.967
P1 (deformation) P1 (accFe)	(M) 0.412	(N) 0.412	(O) 0.412	(P) 0.717

Figure 9-35 – Overall utility values achieved for the different combinations of interaction patterns

The overall utility at a decision point reflects how good a design is, and is a direct reflection of the quality of design outcome (that in turn depend on the design process followed). Based on the achieved utility values shown in Figure 9-35, we observe that:

1. The utility values increase as the interaction patterns are improved from independent to sequential to coupled. This indicates that by introducing

complete information flows between decisions and models, the quality of the final design is better. For example, in the case of a fixed model interaction pattern (P2 for deformation and P2 for accumulated iron), the utility of pattern P6 (0.996) in combination D is better than the utility (0.941) for sequential interaction pattern P5 with product decision made before material decision (combination C), which is in turn higher than the utility achieved using independent interaction pattern P4 (combination A). Similar trend is observed by fixing the decision interaction pattern and varying the model interaction pattern. This is again an intuitive result – better design processes should result in better designs.

2. The maximum utility is equal to 0.996 and is achieved when the decisions are modeled using coupled interaction patterns (P6). The minimum utility of 0.412 is achieved when both decisions and simulation models are modeled using independent interaction pattern (P4 and P1 respectively).
3. The decisions made using sequential decision pattern where the material decision is made before product decision (second column in the matrix) results in the same overall utility as the case where decisions are made independently (first column in the matrix). Such a trend is attributed to the fact that there are four design variables related to the material and only one design variable corresponding to the product. If the material decision is made first, then most of the design freedom is locked. In this specific case, the effect of material design variables on overall utility is significantly greater than the effect of product design variables. Hence, there is not a significant difference

between making the product decision independently or with knowledge about material parameters. It is important to note that this trend is valid only in the context of design problem and preferences formulated in Section 9.3. By changing the preferences, design variables, or their ranges may change this observed trend.

4. Based on the first three points, we conclude that the selection of decision interaction pattern is highly dependent on the model interaction pattern used. This is mainly because the type of model interaction pattern used determines the information about interdependencies captured between parameters. It is the interdependencies between parameters that make the decisions coupled, sequential or dependent.
5. If a sequential pattern is used for calculating deformation, decoupling decisions has little impact on the quality of final outcome as compared to the decoupling of simulation models. This is apparent from the fact that the utility values in a given row are close to each other, but they vary significantly in a given column. This implies if a sequential pattern is used for calculating deformation, the simplest pattern can be used for decisions.
6. Based on the values of overall utility obtained in all the combinations, the combinations A, B, C, D, and L result in an overall utility of 0.900 or above. Hence, these combinations are considered good alternatives for the design process decision. The combinations E, F, G, H, I, J, K, and P result in final designs with overall utility values better than 0.65 and the combinations M, N, and O result in an overall utility value of around 0.41.

As a summary, both ex-post value and the overall utility indicate the appropriateness of interaction patterns. Hence, both these metrics should be considered for making design process decisions. Based on the minimization of ex-post value, the best process options (i.e., the combinations of interaction patterns for decisions and models) include B, C, D, and L. Based on the maximization of overall utility, the best process options include A, B, C, D, and L. The common set of process options using both criteria are – B, C, D, and L. The meanings of these process options are -

B: Sequential decision with material decision made before the product level decision; sequential interaction patterns for both deformation and accumulated iron

C = Sequential decision with product decision made before the material level decision; sequential interaction patterns for both deformation and accumulated iron

D = Coupled material and product decisions; sequential interaction patterns for both deformation and accumulated iron

L = Coupled material and product decisions; sequential interaction pattern for accumulated iron and an independent interaction pattern for deformation.

Note that these process level decisions are also dependent on the time taken to execute the design process. Time for execution of design process has not been included in this study. The results of this section can be extended by include time considerations by including utility functions for time during the calculation of overall utility value.

Limitations of the proposed approach

Further, the reader is cautioned that in this section, the results are presented as if the information about all the interaction patterns is available all at once; and the decision is made about the interaction patterns with the knowledge relating to the outcome from all process options. This approach is adopted in this section to illustrate the tradeoffs between simplification and the quality of decisions made. However, in a real design scenario, the designer starts with a simple interaction pattern, calculate the overall utility and the maximum ex-post value for that decision. Based on these two values, and the resources available to improve the design process, he/she may decide to use that design process option or to use the current process option. The approach presented in this dissertation helps designers to make conscious decisions about improvement of design process. Note that when the designer is utilizing a particular process option and he/she is not aware of the performance of other options, the metrics guide the designer whether improvement in the process is necessary or not. They do not provide any guidance in terms of *how much improvement/refinement is necessary.* For example, based on the available information about the simplest interaction pattern combination (M), the designers cannot determine whether he/she should choose the combination N, O, J, I, or E. It is only after he/she executes the processes using other combinations that he/she can determine the right level of refinement. This is a limitation of the proposed approach.

9.5.5 Model Refinement

The design process decisions discussed so far in Section 9.5 are related to considering the coupling between the design decisions and the simulation models. The meta-level decisions about the level of refinement of each simulation model are not considered. Examples of such refinement decisions include identification of physical phenomena to be considered in a simulation model, level of discretizations in finite element models, invocation of different assumptions in the models, level of detail of boundary conditions to be applied, geometric and parametric idealizations, etc. All these decisions are generally based on the analyst's experience, available computational resources, or the mathematical formalization available. The metric commonly used for evaluating the appropriateness of a simulation model is error in the model's output.

In this dissertation, we advocate the use of value of information metric to quantify the effect of refinement of simulation models on design decisions. This is based on the observation in pressure vessel design example (Section 4.4) and datacenter design example (Chapter 5) that the error is not the only criterion for determining the level of refinement of models. The main criterion is the effect on the quality of decisions, which is measured in terms of the overall payoff. The refinement of simulation models is considered separately to reduce the complexity of design problem under consideration. A separate section is devoted to validate the use of value of information metric for determining the level of refinement of simulation models. In Section 9.6, the design decisions are considered along with the refinement decisions. A design sub-problem from the materials design domain is formulated and executed. The main difference between refinement problems presented in the context of pressure vessel design example in Section 4.4 and that presented in Section 9.6 is that in the pressure vessel design problem, the refinement is considered only in one dimension (i.e., only one parameter is refined at a time), whereas in the materials design problem, two dimensions of refinement are considered simultaneously. Further, the pressure vessel design consists only of in the input variables, whereas the uncertainty in materials design case is due to both imprecision and variability.

9.6 Integrated Model and Design Refinement using Value of Information

The objective in this section is to investigate and validate the use of value of information metric for determining the appropriate level of refinement of the simulation models. The general idea of refinement using value of information is discussed in Chapter 4 and demonstrated using a simple pressure vessel design problem where the material properties are predicted with imprecision bounds using a simulation model. The range of imprecision in material properties predicted by the simulation model can be reduced by refining the simulation model. Hence, the level of refinement of material simulation model in that case is equivalent to determining the largest range that results in good enough decisions.

In this section, we use the same approach to determine the right level of refinement of particle shock simulation model. Although there are various avenues for refinement of shock simulation model, we focus only on refinement via increasing the size of statistical volume element (SVE) and increasing the number of elements in the mesh. Referring back to Section 9.4.1, where the shock simulation model is introduced, the SVE represents a small section of the material through which a shock is propagated. The material morphology in the SVE is randomly generated based on the statistical properties of the distribution of both size of particles, and the distance between them. Since the particles are randomly distributed, the material morphology is different every time a new set of particles is generated, even for the same set of parameters. Hence, the outputs of the simulation models are also randomly distributed. The *size of the SVE* chosen is one of the main factors determining the variability of response. Smaller SVEs have more variability as compared to the larger SVEs. As the size of the SVEs increases, the variability reduces because of the ‘averaging effect’. After the material morphology is generated, the simulation is deterministic i.e., the same morphology with the same boundary conditions will result in the same values for the output parameters. However, since the particle shock simulation is an FEM based simulation model, there is imprecision associated with the outputs due to discretizations. The parameter that can be used to control uncertainty due to discretization is the *number of elements* in the 2-D mesh. The two parameters – size of SVE and the number of elements are model refinement parameters used in this section. The objective is to determine the appropriate values of these parameters in association with the material design parameters. The problem setup for refinement of shock simulation model is discussed in Section 9.6.1. The response surfaces and the associated error calculations are disused in Section 9.6.2. Finally, the results of refinement are presented in Section 9.6.3.

9.6.1 Problem Setup – Refinement of Shock Simulation Model

The problem in this section is to determine the values of two material parameters – size (radius) of aluminum particles and the volume fraction of voids. The range of radius of aluminum particles considered is [0.0005 0.0015]mm and the range of volume fraction of voids is [0.02 0.10]. All other parameters related to the material properties are assumed constant. In addition to determining the material parameters, the objective is also to determine the appropriate values for two model parameters – size of SVE and number of elements in the mesh. The size of SVE is a function of the length of the SVE, which lies within [0.014 0.028]mm. The width is taken as half of the length in order to maintain the same aspect ratio for all cases. The number of elements in the mesh vary from [200 400]. Note that the decisions about the material parameters and the simulation model parameters depend on each other. Depending on the preferences for material properties, the appropriate level of uncertainty in simulation models may change. Similarly, depending on the level of model refinement chosen, the decision about material properties may change. The two decisions and their dependency are shown in Figure 9-36. As the simulation model is refined by a) increasing the size of SVE, and b) increasing the number of elements in the mesh, the complexity of the model and associated runtime also increases significantly. In order to make efficient decisions, an appropriate set of values for the model related parameters is desired. In this section, we discuss how to quantify this tradeoff and make decisions.

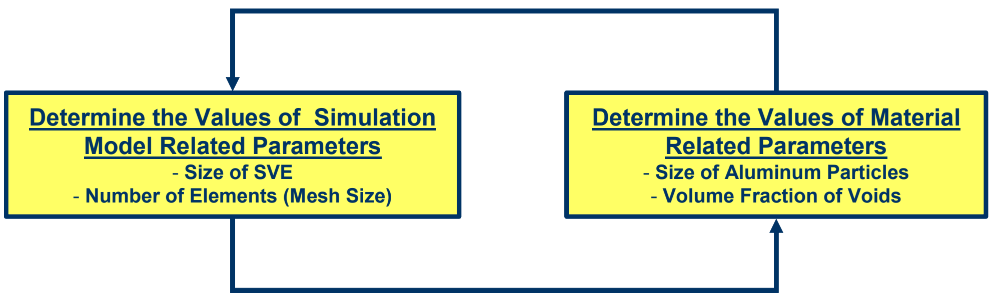


Figure 9-36 – Determination of simulation model and material parameters in an integrated fashion

The objective is to satisfy goals for the shock wave velocity with low variance. Shock speed is chosen as an objective because the material properties depend directly on the shock speed. As discussed previously in this chapter, the material properties can be modeled using the Mie-Gruneisen Equation of State (EOS) which is determined by fitting a straight line on the Particle speed – Shock speed data. Hence, the material properties are dependent on the shock speed achieved for a given particle speed. In the problem under consideration, the particle speed is fixed to 1000m/sec and the shock speed is calculated using the particle shock simulation model. Since we are dealing with the case where simulation model is approximate and the outputs consist of uncertainty, the objective is formulated with consideration of robustness. The shock speed goal is divided into two sub-goals – bringing the mean to target, and minimization of variation around the mean. This variation is a combination of the effects of both variation in the response due to changing morphology and the imprecision due to discretizations. The preferences for mean and variance are modeled as individual utility functions that are combined together as linear weighted functions.

A Window Size = 0.014 Num Elements = 200	B Window Size = 0.021 Num Elements = 200	C Window Size = 0.028 Num Elements = 200
D Window Size = 0.014 Num Elements = 300	E Window Size = 0.021 Num Elements = 300	F Window Size = 0.028 Num Elements = 300
G Window Size = 0.014 Num Elements = 400	H Window Size = 0.021 Num Elements = 400	I Window Size = 0.028 Num Elements = 400

Figure 9-37 - Nine options for refinement explored for the shock simulation model

The size of SVE (also referred to as window size) and the number of elements in the mesh can be varied continuously between the lower and upper bounds, which provides an infinite set of options of simulation model refinements. However, exploring all those options is not effective from a decision making perspective. In order to reduce the computational load, we just explore nine different simulation model refinement options. These options are generated by taking all combinations of three levels each of size of SVE (with lengths 0.014, 0.021, and 0.028mm) and the number of elements in the mesh (200, 300, and 400 along the x-axis with half that number in the y-direction). The nine options are labeled from A through I. The approach is to select the simplest model (window size = 0.014, and number of elements = 200) and sequentially make it more detailed.

9.6.2 Simulation Runs and Response Surface Model Development

The particle shock simulation is a computationally expensive model, whose complexity increases with the increase in mesh density and window size. Performing design exploration using the simulation model directly is difficult. Hence, instead of the simulation models, we rely on using response surfaces. The first response surface is of the shock wave speed as a function of the design variables (size of aluminum particles and volume fraction of voids) and the window size. In this first response surface, the mesh size is assumed constant.

Table 9-10 - Design of experiments table for radius of aluminum particles, volume fraction of voids, and the window size

Run No.	Radius of Aluminum Particles (mm)	Volume Fraction of Voids	Length of SVE (window size)
1.	0.0005	0.02	0.014
2.	0.0015	0.02	0.014
3.	0.0005	0.10	0.014
4.	0.0015	0.10	0.014
5.	0.0005	0.02	0.028
6.	0.0015	0.02	0.028
7.	0.0005	0.10	0.028
8.	0.0015	0.10	0.028
9.	0.0010	0.06	0.021
10.	0.0005	0.06	0.021
11.	0.0015	0.06	0.021
12.	0.0010	0.02	0.021
13.	0.0010	0.10	0.021
14.	0.0010	0.06	0.014
15.	0.0010	0.06	0.028

The reason for keeping the mesh size constant is to decouple the effect of variability due to changing morphology and the error due to mesh size. Central Composite design of experiments is used for performing experiments and generating the data for response surface. The design of experiments table is shown in Table 9-10. In order to quantify the effect of variability, the simulation is carried out 20 times with changing material morphology.

A response surface is fit based on the data generated at these 15 points with 20 replicates. The response surface plots for the shock wave speed as a function of size of aluminum, volume fraction of voids and window size are shown in Figure 9-38. The corresponding residual plots are shown in Figure 9-39.

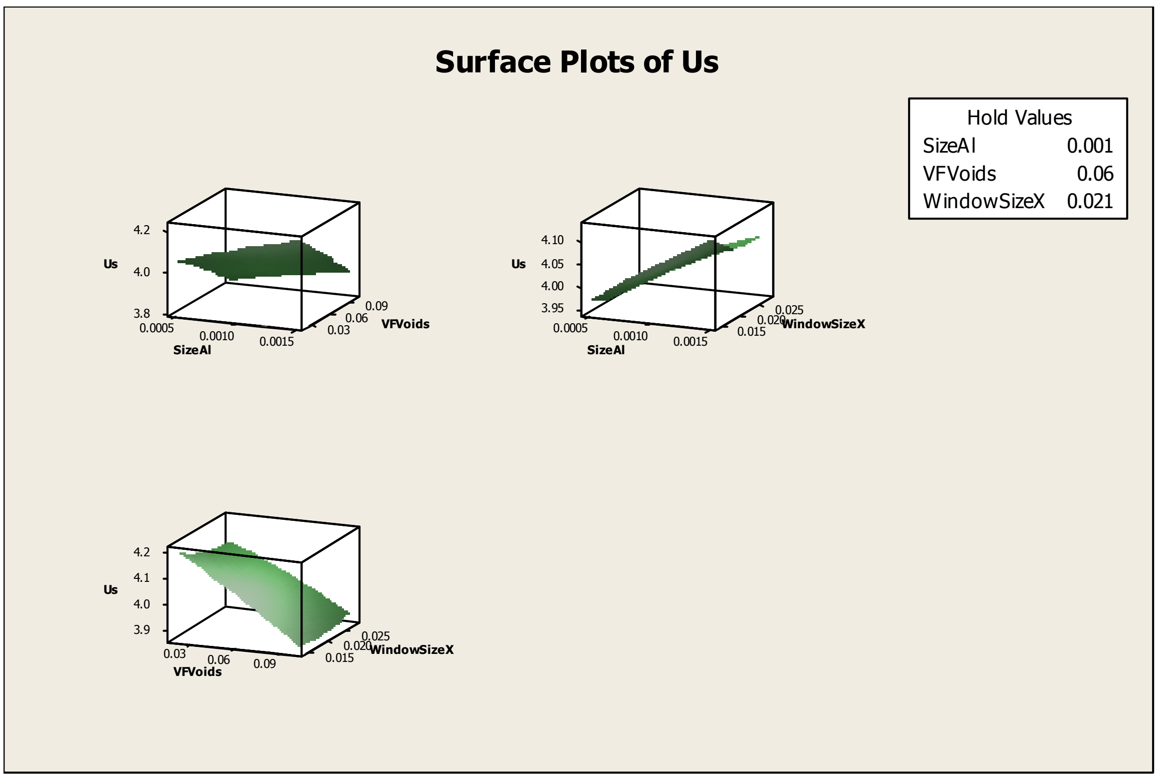


Figure 9-38 - Response Surface – Us as a function of SizeAl, VFVoids, WindowSizeX

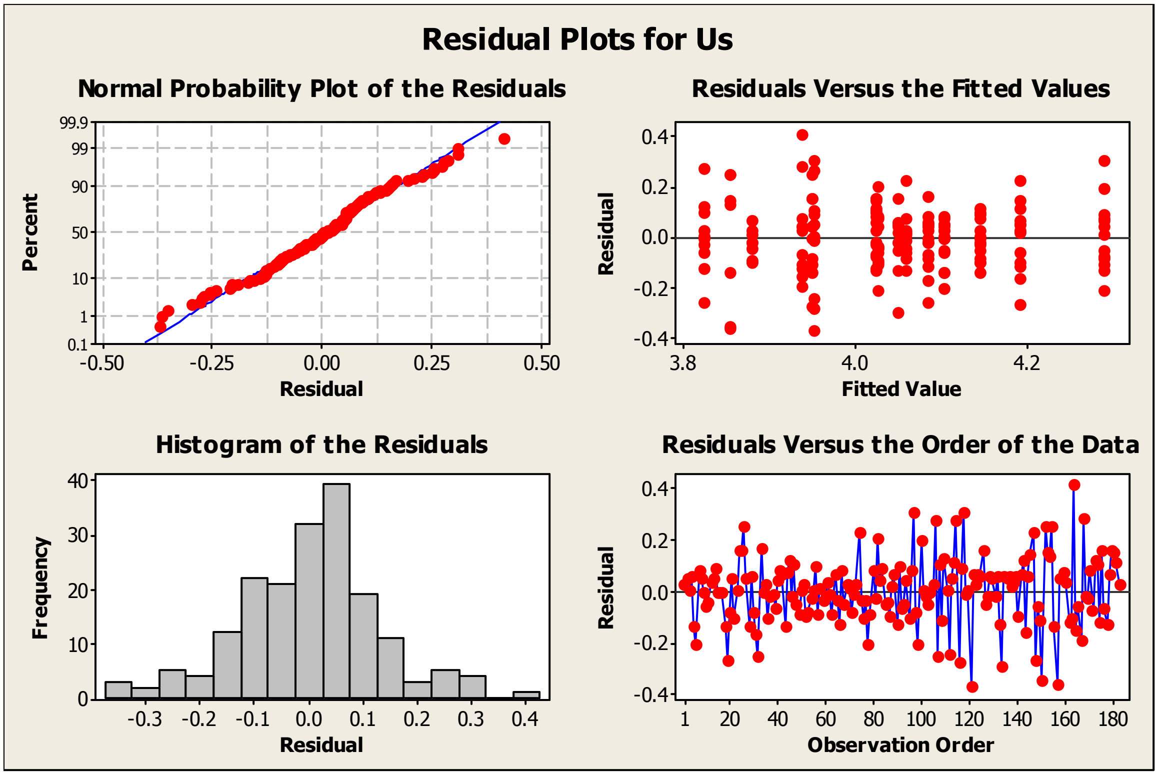


Figure 9-39 - Residual plot for shock wave velocity (Us)

In addition to the mean values of the response variable (shock wave speed), a response surface is fit for the variance in output at each point in the design space. For each data point in Table 9-10, the variance is calculated and a response surface is fit between the variance of shock wave speed and the three variables. The third response surface is for capturing the effect of changing mesh size on the output of simulation model. The data generated for changing mesh size is based on the same material morphology. In this case, the window size is fixed to 0.028mm. Since the simulation model is deterministic after the initial material morphology is generated, the simulation model is run only once for a combination of size of aluminum particles, volume fraction of voids, and the mesh size.

Three different response surfaces are fit between the response (shock wave speed) and design variables (size of aluminum and volume fraction of voids) for three mesh sizes of 400, 300, and 200 respectively. Recall that the numbers 200-400 represent the number of elements along the x-direction. The number of elements along the y-direction is half of the number of elements in x-direction. Hence, if the mesh size is 400, then the total number of elements in the 2-D SVE is 400*(400/2) = 80000. It is assumed that the simulation with a mesh size of 400 is the most accurate and the error at a smaller mesh size is calculated by taking the absolute value of difference between the two values obtained by corresponding response surfaces. The response surfaces for mesh sizes of 400, 300, and 200 are plotted in Figure 9-40, Figure 9-41, and Figure 9-42 respectively.

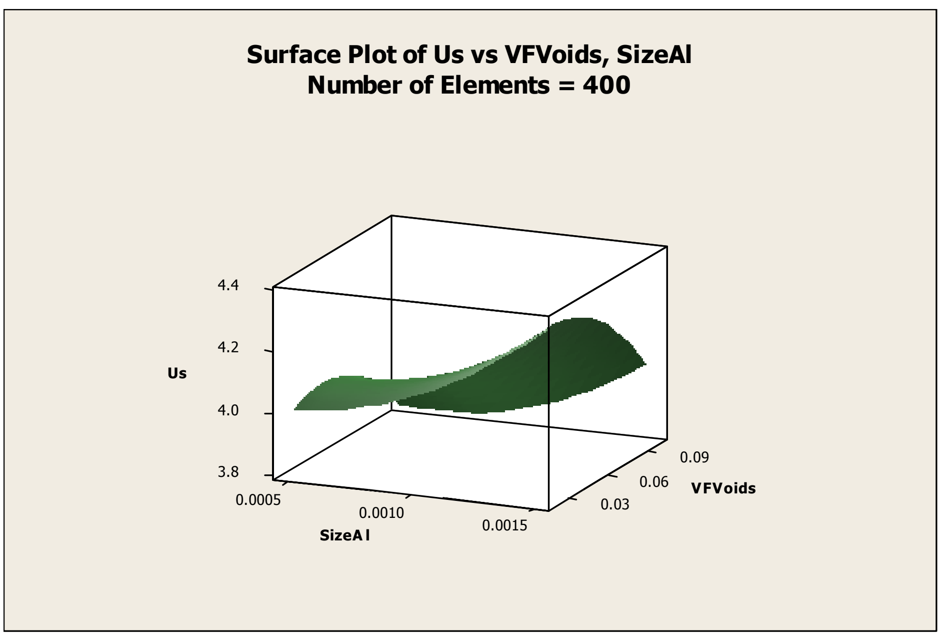


Figure 9-40 - Response surface for shock wave speed at a mesh size of 400

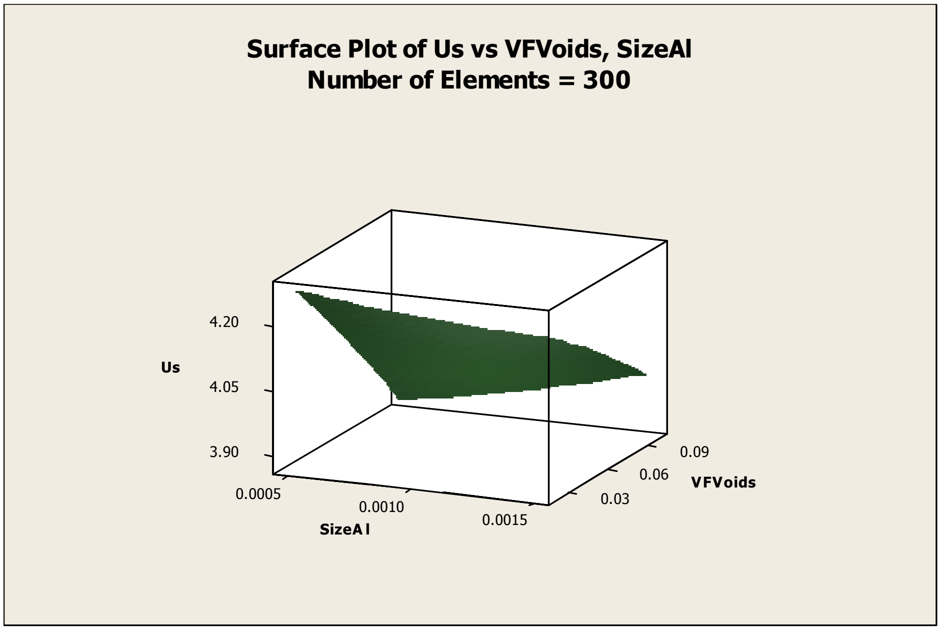


Figure 9-41 - Response surface for shock wave speed at a mesh size of 300

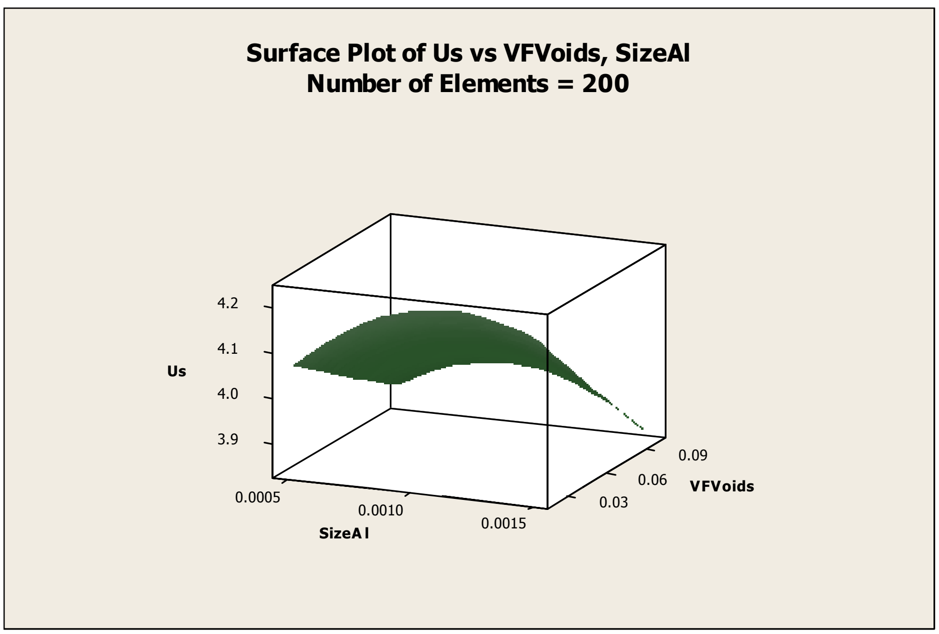


Figure 9-42 - Response surface for shock wave speed at a mesh size of 200

The total variation in the shock wave speed is a combination of the variation due to changing morphology and the error due to consideration of a coarse mesh. The calculation of variation of response due to changing material morphology is shown earlier in this section. The calculation of error due to a coarse mesh is based on the assumption that the mesh size of 400 is the most accurate based on the available data. This assumption is just to characterize the error in the simulation model. If true experimental data were available, that would be used for characterization of the error in the model. Using the information at a mesh size of 400, the error at mesh size of 300 and 200 is calculated as the absolute value of the difference in shock wave speed. The total deviation in response is equal to the sum of variation and the imprecision. The mean value is based on the average shock wave speed calculated using the finest mesh (mesh size of 400). The mean, lower and upper bound of the shock wave speed as a function of design variables (size of aluminum and volume fraction of voids) is plotted for different values of mesh size and areas of SVE in Figure 9-43. This completes the characterization of the simulation models. The next step is to model the preferences in terms of utility functions.

As mentioned previously in this section, the problem is formulated with two goals – achievement of target of shock wave speed and minimization of its variance due to imprecision and uncertainty. The preferences are modeled as risk-averse utility functions. The plot of utility as a function of mean and variance of shock speed is shown in Figure 9-44. It is important to note that changing the utility function will have effect on the specific decisions (values of design variables and level of refinement chosen) made using the method followed. However, it does not affect the steps followed in the method. Even by changing the shape of the preferences, the conclusions about the applicability of the method and the metrics derived from the method remain the same.

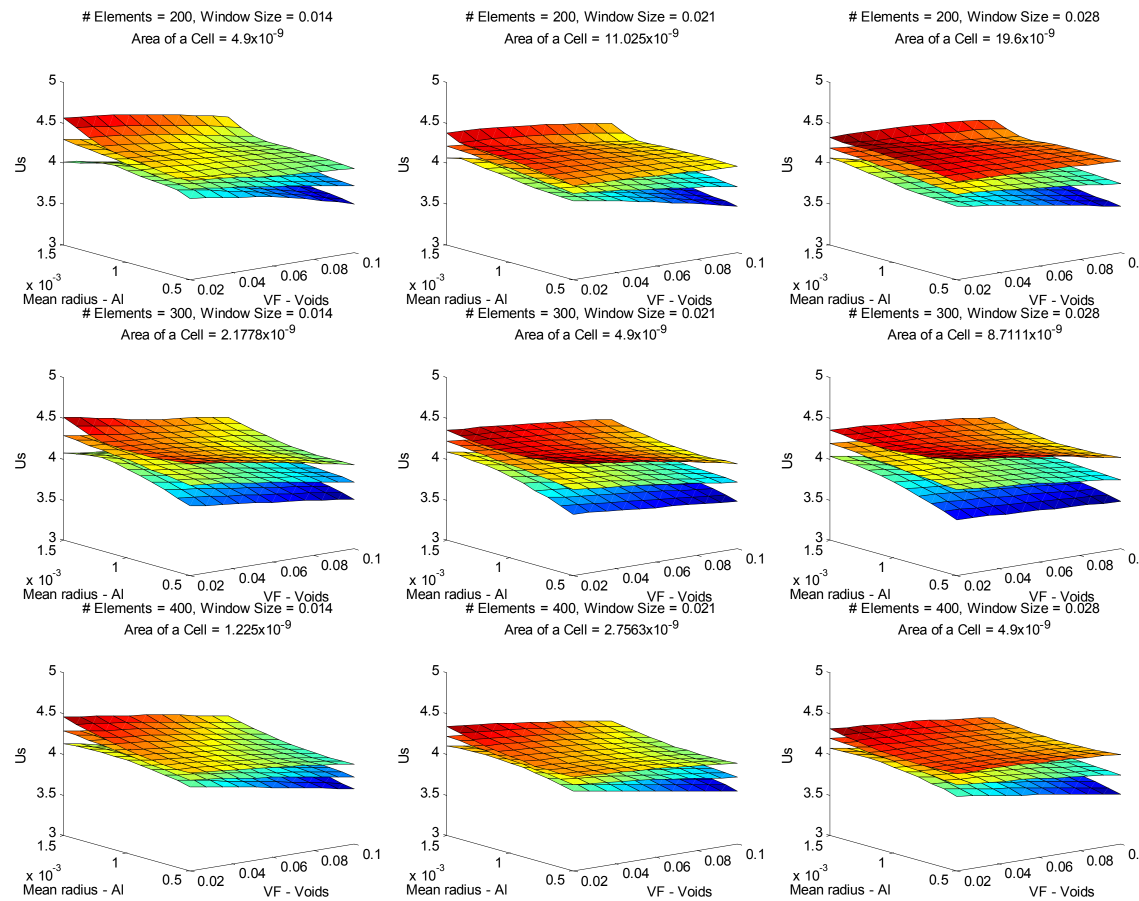


Figure 9-43 - Surface plots of shock speed as a function of design variables for different refinement scenarios

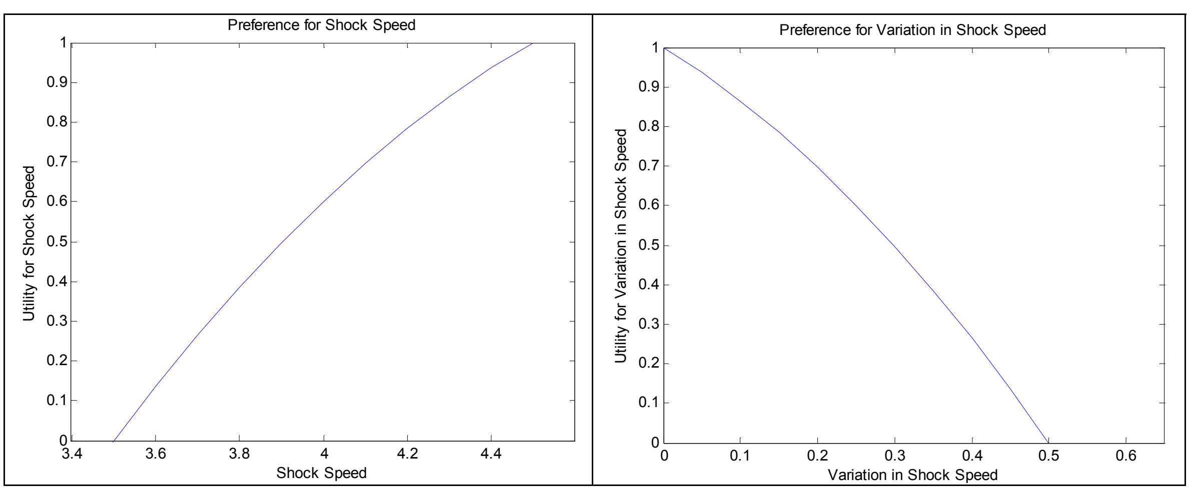


Figure 9-44 - Utility functions for shock wave speed and the variation in shock wave speed

9.6.3 Results from Refinement of Shock Simulation Model Using Value of Information

The decisions made using the different levels of refinement of the simulation model are presented in Table 9-11. The results are shown for three different cases with different weights assigned to the utility for mean shock wave speed and the deviation in shock wave speed during the calculation of overall utility. These three cases refer to different preference conditions. In Case 1, the weight for both mean and deviation is equal to 0.5. In case 2, the weight for mean is 1.0 and the weight for preference is 0.0. Finally in Case 3, the weight for mean is 0.0 and the weight for deviation is 1.0. The results in Table 9-11 contain columns for the number of elements in the mesh and the window size that determine the level of refinement of the shock simulation model. The third column is for the area of each cell in the finite element model and the fourth column is for maximum overall utility achieved at the decision point. The following two columns are for design variables – mean size of aluminum and volume fraction of voids. The corresponding values of response variables - mean shock wave speed and the variation in shock wave speed is presented in the next two columns. Finally, the last column indicates the ex-post range of utility, which is the difference between the maximum overall utility that can be achieved at any point in the design space and the minimum utility achieved at the decision point. This ex-post range is used as a metric for value of information that can be achieved by refining the simulation model further. If the ex-post value is low enough, the model does not need to be refined. If the ex-post value is high, there is scope for improving the decision via refinement of the simulation model.

Table 9-11 - Comparison of decisions made using different levels of refinement of simulation model using the upper bound on ex-post value

Num Elements	Window Size	Area of Cell (e-9)	Max Overall Utility	Mean Al Size	VF Void	Shock Speed	Variation Shock Speed	Ex-Post MaxUtility	Minimum Utility At Decision Point	Ex-Post Range (Value of Information)
CASE 1: Weight for mean = 0.5, weight for deviation = 0.5										
200	0.014	4.9	0.847	0.0011	0.02	4.222	0.081	1.000	0.733	0.267
200	0.021	11.0	0.830	0.0014	0.02	4.207	0.096	0.922	0.706	0.216
200	0.028	19.6	0.799	0.0014	0.02	4.182	0.122	0.884	0.658	0.226
300	0.014	2.2	0.845	0.0012	0.02	4.240	0.094	1.000	0.737	0.263
300	0.021	4.9	0.811	0.0014	0.02	4.207	0.120	0.912	0.684	0.228
300	0.028	8.7	0.772	0.0015	0.02	4.195	0.161	0.947	0.633	0.314
400	0.014	1.2	0.851	0.001	0.02	4.204	0.065	0.973	0.731	0.242
400	0.021	2.8	0.845	0.0011	0.02	4.160	0.047	0.892	0.708	0.184
400	0.028	4.9	0.823	0.0012	0.02	4.154	0.075	0.874	0.677	0.197
CASE 2: Weight for mean = 1.0, weight for deviation = 0.0										
200	0.014	4.9	0.857	0.0015	0.02	4.291	0.275	1.000	0.616	0.384
200	0.021	11.0	0.802	0.0015	0.02	4.222	0.157	0.922	0.663	0.259
200	0.028	19.6	0.780	0.0015	0.02	4.195	0.131	0.884	0.662	0.221
300	0.014	2.2	0.857	0.0015	0.02	4.291	0.220	1.000	0.669	0.331
300	0.021	4.9	0.802	0.0015	0.02	4.222	0.128	0.912	0.690	0.222
300	0.028	8.7	0.780	0.0015	0.02	4.195	0.161	0.947	0.633	0.314
400	0.014	1.2	0.857	0.0015	0.02	4.291	0.165	0.973	0.720	0.253
400	0.021	2.8	0.802	0.0015	0.02	4.222	0.116	0.892	0.702	0.190
400	0.028	4.9	0.780	0.0015	0.02	4.195	0.119	0.874	0.674	0.200
CASE 3: Weight for mean = 0.0, weight for deviation = 1.0										
200	0.014	4.9	0.908	0.0007	0.02	4.148	0.070	1.000	0.675	0.325
200	0.021	11.0	0.870	0.0014	0.02	4.207	0.096	0.922	0.706	0.216
200	0.028	19.6	0.837	0.0014	0.036	4.140	0.118	0.884	0.622	0.261
300	0.014	2.2	0.873	0.0012	0.02	4.240	0.094	1.000	0.737	0.263
300	0.021	4.9	0.836	0.0013	0.02	4.192	0.118	0.912	0.671	0.240
300	0.028	8.7	0.774	0.0014	0.02	4.182	0.156	0.947	0.626	0.322
400	0.014	1.2	0.925	0.0008	0.02	4.167	0.058	0.973	0.704	0.269
400	0.021	2.8	0.949	0.0009	0.02	4.127	0.040	0.892	0.684	0.208
400	0.028	4.9	0.909	0.001	0.02	4.126	0.069	0.874	0.655	0.219

The ex-post range for Case 1 is presented in a matrix form in Figure 9-45, where the rows correspond to fixed values of number of elements and the columns correspond to fixed values of the window size. The values in the cells for Ex-Post range or Value of Information (VOI) are presented in each of the cells in the matrix. There are two dimensions along which refinement can take place. The simulation model can be refined by either increasing the window size, or by increasing the number of elements. As the simulation model is refined, the value for VOI metric reduces because it means that the possible improvement in the solution by refining the model also reduces. This is observed in Figure 9-45 while going from A→B (increasing the window size) or from A→D (increasing the number of elements). The ex-post value reduces from 0.267 to 0.216 by refinining the model from A→B and the value reduces from 0.267 to 0.263 by refining the model from A→D. This is an expected result.

A Window Size = 0.014 Num Elements = 200 VOI = 0.267	B Window Size = 0.021 Num Elements = 200 VOI = 0.216	C Window Size = 0.028 Num Elements = 200 VOI = 0.226
D Window Size = 0.014 Num Elements = 300 VOI = 0.263	E Window Size = 0.021 Num Elements = 300 VOI = 0.228	F Window Size = 0.028 Num Elements = 300 VOI = 0.314
G Window Size = 0.014 Num Elements = 400 VOI = 0.242	H Window Size = 0.021 Num Elements = 400 VOI = 0.184	I Window Size = 0.028 Num Elements = 400 VOI = 0.197

Figure 9-45 - Refinement Case 1: Weight for mean shock wave speed = 0.5, weight for variation in shock wave speed = 0.5

When the window size is increased from 0.021 to 0.028 from B→C, while keeping the same number of elements (=200), the ex-post value actually increases from 0.216 to 0.226. This is opposite to the expected trend. The reason for that is that the two ways of refinement of the simulation model, increasing the window size and increasing the number of elements, are not independent of each other. By increasing the area of the SVE and keeping the same number of elements, the area of each cell actually increases. The area of each cell in B is equal to 11.05×10^{-9} mm^2, whereas the area of each cell in C is equal to 19.6×10^{-9} mm^2. Hence, although the variability reduces due to increase in the area of the SVE, the imprecision increases due to increase in the area of cell. Similar trend is observed while going from E→F. A comparison of the VOI metric for all the refinement levels A through I indicate that if the refinement decision is only based on the lowest ex-post value, then the model H with window size 0.021 and number of elements equal to 400 is the best because it gives the minimum ex-post value (=0.184). This model corresponds to a low area of cell (2.7505×10^{-9} mm^2) and a large number of elements

(400). It is important to note however, that as the number of elements or the window size is increased the computational cost also increases. The cost of computation is not considered in the determination of best model. The cost can be considered during the calculation of overall utility function.

A Window Size = 0.014 Num Elements = 200 VOI = 0.384	B Window Size = 0.021 Num Elements = 200 VOI = 0.259	C Window Size = 0.028 Num Elements = 200 VOI = 0.221
D Window Size = 0.014 Num Elements = 300 VOI = 0.331	E Window Size = 0.021 Num Elements = 300 VOI = 0.222	F Window Size = 0.028 Num Elements = 300 VOI = 0.314
G Window Size = 0.014 Num Elements = 400 VOI =0.253	H Window Size = 0.021 Num Elements = 400 VOI = 0.190	I Window Size = 0.028 Num Elements = 400 VOI = 0.200

Figure 9-46 - Refinement Case 2: Weight for mean shock wave speed = 1.0, weight for variation in shock wave speed = 0.0

A Window Size = 0.014 Num Elements = 200 VOI = 0.325	B Window Size = 0.021 Num Elements = 200 VOI = 0.216	C Window Size = 0.028 Num Elements = 200 VOI = 0.261
D Window Size = 0.014 Num Elements = 300 VOI = 0.263	E Window Size = 0.021 Num Elements = 300 VOI = 0.240	F Window Size = 0.028 Num Elements = 300 VOI = 0.322
G Window Size = 0.014 Num Elements = 400 VOI = 0.269	H Window Size = 0.021 Num Elements = 400 VOI = 0.208	I Window Size = 0.028 Num Elements = 400 VOI = 0.209

Figure 9-47 - Refinement Case 3: Weight for mean shock wave speed = 0.0, weight for variation in shock wave speed = 1.0

Similar results are shown for other preference cases in Figure 9-46 and Figure 9-47. It is observed that in all the cases, the model with the lowest ex-post value is model H. This is just a characteristic of this problem. The minimum could lie with different models in which case different models would be appropriate for different preference conditions as shown in the pressure vessel design in Section 4.4. The value of VOI is different in different cases. In this section, we provide the results for all the refinement cases right away. However, the designers would actually select the simplest model first and then refine it sequentially until a good enough model which satisfies a cutoff value of the value of information is realized. For example, if a cutoff value of 0.200 is specified prior to refinement of simulation models, the designer would select model I in Cases 1 and 2. None of the models A through I would be suitable in Case 3 because there is no model that satisfies the cutoff for ex-post value. The designer would have to refine the model further.

9.7 Verification and Validation

In this chapter, two aspects of the validation of the framework are addressed – Empirical Structural Validation and Empirical Performance Validation which are discussed in Sections 9.7.1 and 9.7.2 respectively. A visual overview of the aspects of validation square addressed in this chapter is provided in Figure 9-48. The validation square provides Chapter 9 specific details to the overall validation square presented in Figure 1-13.

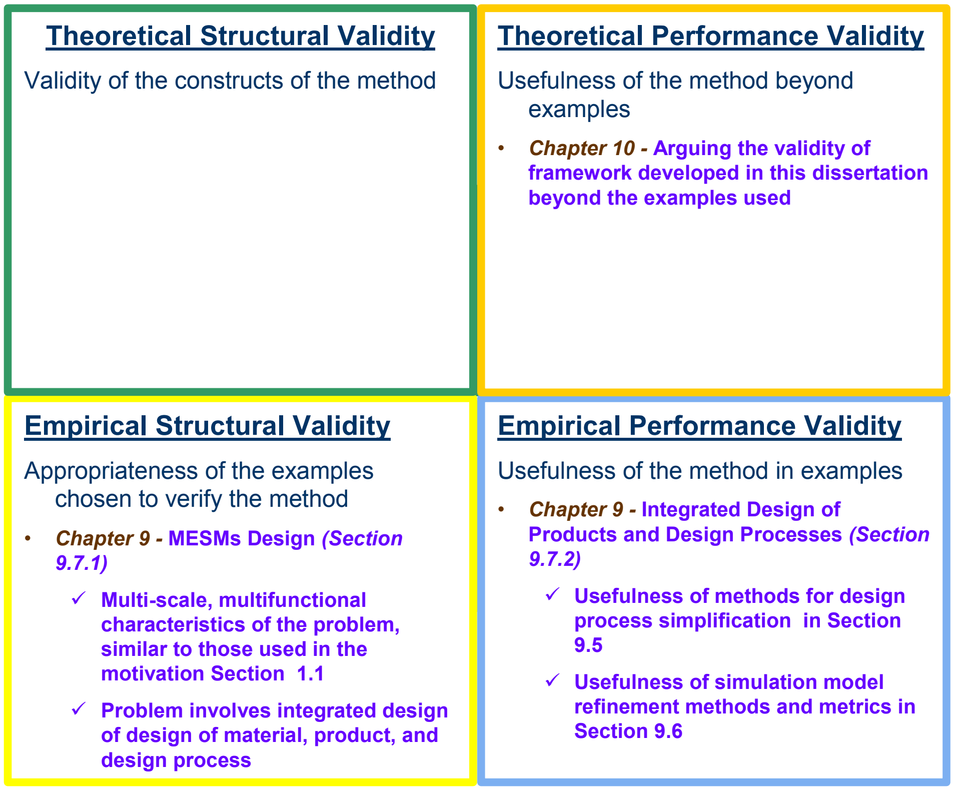


Figure 9-48 – Aspects of validation of the design framework addressed in Chapter 9

9.7.1 Empirical Structural Validation

Empirical structural validation involves accepting the appropriateness of the example problems used to verify the performance of the method. The example discussed in this chapter is a multiscale, multifunctional design problem similar to the one discussed in the motivation Section 1.1. The problem involves multiple simulation models and decisions at different scales that are coupled with each other. The information about information flow between simulation models is available. Further, the problem consists of decisions related to three aspects: design of material, product, and the design process. All the three decisions depend on each other and ultimately affect the final product performance. We believe that this is a reasonably complex multiscale design scenario. Further, simulation models are available at different scales for the materials design scenario. The design process can be represented in terms of decisions that can be mathematically formulated and are supported using available simulation models. Information about error characterization for simulation models is available from the response surfaces. Hence, the materials design problem contains all the required ingredients for validating the design methods for simplification of design processes presented in the Sections 5.3.1 and 5.4.1.

The particle shock simulation model is associated with uncertainty both due to statistical variability and imprecision. The computational time for execution of the model depends significantly on the size of SVE and mesh size chosen. This motivates the need for selecting the appropriate level of refinement of the model such that there is a balance between impact on decision and the cost of executing the model. Hence, the shock-simulation model is appropriate for the validation of value of information metric for model refinement presented in Section 4.3.

9.7.2 Empirical Performance Validation

Empirical performance validation consists of accepting the usefulness of the outcome with respect to the initial purpose and accepting that the achieved usefulness is related to applying the method. The empirical performance validation in this chapter is carried out in two phases – the validation of design process simplification in Section 9.5 and simulation model refinement in Section 9.6. In Section 9.5, the interaction patterns proposed in Section 3.5.2 are used to model the flow of information between design decisions and simulation models. The interaction patterns are used to simplify the design processes by simplifying the interactions between models and decisions. It is shown that by using the simplification methods presented in Sections 5.3.1 and 5.4.1, the combinations of simple interaction patterns – B, C, and L are identified. These combinations result in similar design decision as the most complex interaction pattern -

D. Hence, the efficiency of design decision making can be increased without affecting the quality of decisions. The value of information metric is also shown to be effective for making such process level decisions. The application of the value of information metrics for refinement of simulation models is shown in Section 9.6. It is shown that the metric is valuable for determining the right level of detail in simulation model for design decision making. It is also shown that the metric captures the effect of both uncertainty in the simulation models and the designers' preferences. Based on the results, it is argued that the metric is appropriate for determining the level of simplification in multiscale models.

9.8 Role of Chapter 9 in this Dissertation

In this chapter, we present a materials design example for the validation of design methods and metrics presented in Chapter 3, Chapter 4, and Chapter 5. The relationship of the validation example presented in this chapter with other chapters is presented in Figure 9-49. The design problem is modeled as integrated design of materials, products, and design processes. The results from the design problem in this chapter indicate the usefulness of proposed methods and metrics in the context of a multiscale design problem. The following chapter is the closure of this dissertation, where a summary of the research presented in the first nine chapters is presented along with the validation, contributions and avenues for future work.

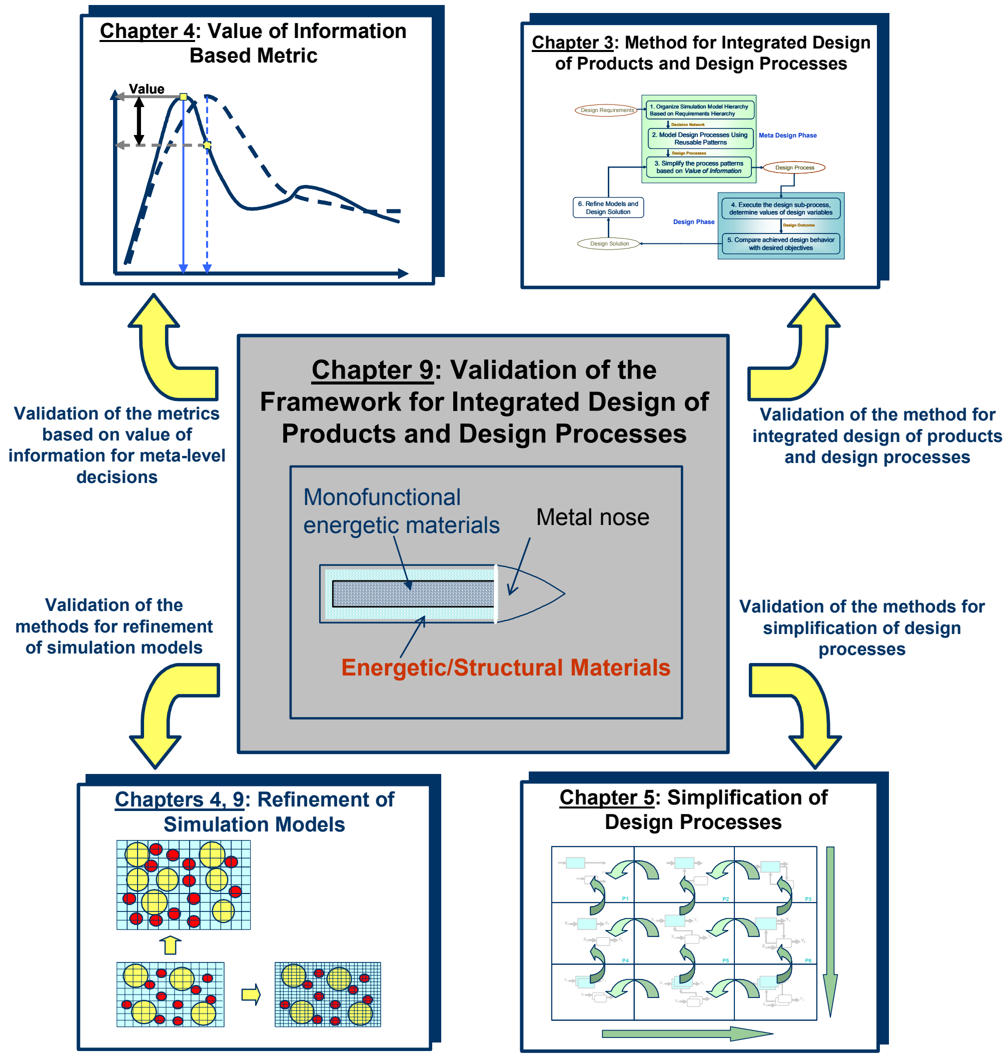


Figure 9-49 – Relationship of Chapter 9 with other chapters in the dissertation

Chapter 10 Closure

10.1 A Summary of the Dissertation

In this dissertation, a framework for the integrated design of products and design processes is established. The framework is established as an embodiment of the primary hypothesis in this dissertation: ***"Simulation-based design of multiscale multifunctional systems can be carried out by decision-based integrated design of products and design processes"***. The hypothesis is used to answer the primary research question for the dissertation: ***"How should simulation-based design of complex multiscale, multifunctional systems be carried out?"***

The framework established in this dissertation consists of three main components: a) a Robust Multiscale Design Exploration Method (RMS-DEM) that consists of three phases – meta-design, design process execution, and refinement, *b)* metrics and methods for simplification of complex design process and simulation model refinement using information economics and robustness, and c) an information modeling strategy for simulation-based design information to support design process exploration and information reuse. Designing design processes is equivalent to making decisions such as:

a) Which physical phenomena should be modeled?,

b) Which couplings are critical from a design standpoint?, *c)* how much refinement in models is necessary?,

d) Do we need additional experimental data?, and

e) What should the sequence of decisions and simulation tasks be?

These design process decisions affect both the design outcome and the efficiency with which the outcome is achieved. Hence, designing design processes is essentially making these meta-level decisions while accounting for the tradeoff between performance of the final product, confidence in design decisions, and resources utilized in design decision-making. The primary research question is partitioned into the following three research questions –

Q1) *How can simulation-based multiscale design processes be designed in association with products?*

Q2) *How should multiscale design processes be systematically simplified and models refined in a targeted manner to support faster design decision making without compromising their quality?*

Q3) *How should simulation-based design processes be modeled in a systematic manner and represented in a computer interpretable format to support design process exploration?*

The context for answering these research questions in the dissertation is design of multiscale systems that are dominated by horizontal couplings (i.e., coupling between physical phenomena at a single scale) and vertical couplings (i.e., coupling across different scales). Due to these couplings, the design processes cannot be derived directly from their functional decomposition, as required by conventional design methods. Hence, the coupling and interactions aspect of designing design processes is explored throughout this dissertation.

In this dissertation, we address the design of design processes from a systems-based perspective. Similar to the manner in which products are designed as systems with subsystems and interfaces, we view design processes as modular systems composed of reusable building blocks. These building blocks are described using the types of interaction and information flow between simulation models and decisions. In general, the interactions between two simulation models or decisions can be independent, sequential, or coupled. Coupled interaction patterns are most complex and the independent interaction patterns are simplest. The interaction patterns are suitable for modeling simulation-based design processes applicable to multiscale design. These building blocks are used to answer the first research question related to design process configuration. After modeling the design processes using the interaction patterns, they can be simplified by ignoring the information flows that do not have a significant on the design. This requires a metric to determine the impact of simplifying an interaction pattern. The metric developed and used in this dissertation is based on value of information. This metric is used to answer the second research question that is related to systematic simplification and targeted refinement of simulation models. The first two research questions tie to the theoretical aspects of the framework and the third research question is related to the implementation of these theoretical concepts in the form of a computational framework that supports both design of products and associated design processes. The approach (3-P) used for answering the third research question is a synthesis of three key components: *a)* a decision-based view of design processes and the adaptation of a specific instantiation, namely the Decision Support Problem (DSP) Technique, *b)* a modular systems based approach for design processes, and *c)* a mechanism for separation of declarative and procedural information.

The framework is demonstrated and validated using various example problems: structure design, pressure vessel design, datacenter design, multifunctional LCA design, and multifunctional energetic structural material design. The structure design example is used to explain the steps of the design method in Chapter 3. The pressure vessel design problem is used to validate the value of information metric in Section 4.4. The datacenter design problem is used to validate the methods for scale and decision decoupling in Sections 5.3.2 and 5.4.2 respectively. The LCA design example is used to validate the interval-based focalization method for multifunctional design in Section 6.4.3. The pressure vessel design example is again used in Section 8.4 to validate the implementation of information modeling strategy. Finally, the multifunctional energetic structural materials design example is used as a comprehensive example throughout Chapter 9 to validate the methods and metrics proposed in the dissertation. The details of validation and answers to the research questions are provided in Section 10.2.

As a summary, ***the primary contribution from this dissertation is systems-based approach for integrated, systems-based design of products and design processes***, Specifically, the contributions include *a)* a design method based on systematic simplification and refinement, *b)* methods for design process simplification, *c)* a set-based method for multifunctional decentralized design, *d)* a set of metrics for design processes based on information economics, *e)* an information modeling strategy to provide computational support for design of products and design processes. The details of the contributions from this dissertation are discussed in Section 10.3.

10.2 Answering the Research Questions and Validating the Hypotheses

The framework for integrated design of products and design processes is established in this dissertation to answer the primary research question – *How should simulation-based design of complex multiscale, multifunctional systems be carried out?* The hypothesis used to answer this question is that *simulation-based design of multiscale, multifunctional systems can be carried out by decision-based integrated design of products along with their design processes*. The framework developed in this dissertation embodies this primary hypothesis and consists of three components. These three components embody the hypotheses associated with three sub-research questions. These hypotheses are proposed in Chapter 1 and are addressed throughout the dissertation. In this section, we revisit those hypotheses and corresponding questions.

For the readers' convenience, a summary of research questions, hypotheses, validation square and contributions are highlighted in Figure 10-1. In this figure, the validation square is divided into five validation sub-squares, each subsquare corresponding to one of the hypothesis H1.1, H1.2, H2.1, H2.2, or H3.1. The validation subsquares are labeled according to the hypothesis to be validated. For example, the validation square corresponding to hypothesis H1.1 is labeled VSQ1.1. Each of these subsquares is discussed in this section to establish the validity of the overall framework presented in this dissertation. Summaries of arguments made throughout the dissertation regarding theoretical and empirical validation for each of the hypotheses are provided in Sections 10.2.1, 10.2.2, and 10.2.3. In Section 10.2.4, attention is devoted to theoretical performance validation, which involves building confidence in the framework presented for scenarios beyond the specific examples chosen for validation.

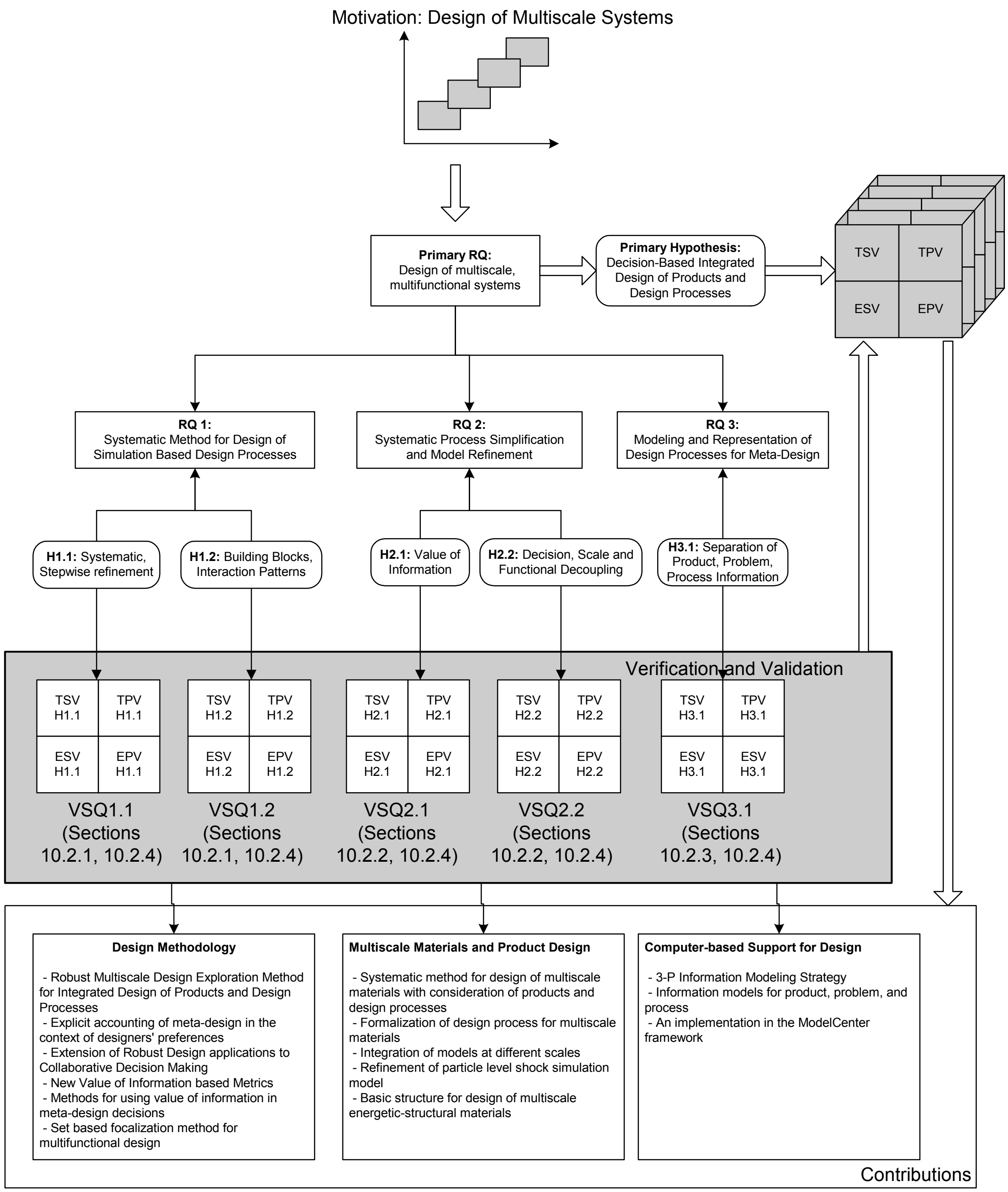


Figure 10-1 – Summary of the dissertation

10.2.1 Research Question 1 – Configuration of Design Processes for Effective Decision Making

Hypothesis H1.1: First research question in this dissertation is related to the efficient configuration of design processes to make efficient design decisions. Two hypotheses are used to answer this research question. The first hypothesis (H1.1) is that *systematic stepwise refinement of design processes and associated products increases the efficiency and effectiveness of design decision making*. This hypothesis is embodied in the robust multiscale design exploration method consisting of two phases – meta-design and design. In the meta-design phase, the design process to be executed is first configured and in the design phase, the design process is executed. A refinement loop around the two phases is used to systematically increase the fidelity of design process elements and the interactions between them, thereby refining the design solution. The strategy advocated in this method is to: *a)* use the simplest design processes and to use the robust design methods to make decisions in the presence of uncertainty as a result of simplification of design process, and then *b)* systematically refine both the simulation models and design processes to refine the design solutions.

The validation square VSQ 1.1 for validating Hypothesis H1.1 is shown in Figure 10-2. VSQ 1.1 is a validation sub-square for the overall dissertation level validation square (see Figure 10-1). Aspects of this figure are addressed in various sections throughout the dissertation. These sections are referenced in the figure. The ***theoretical structural validation*** of Hypothesis 1.1 is carried out by performing a literature review of existing literature focused on improvement or the design of design processes. Examples of literature focused in improvement of design processes include concurrent engineering and Product Lifecycle Management (PLM). Examples of literature focused on designing the design processes include decision-based design (specifically the DSP Technique) and the Design Structure Matrix (DSM). Based on the literature, it is concluded that it is not possible to generate all possible design process alternatives and execute them to evaluate and compare the performance of all processes before selecting the most appropriate design process. This is because execution of all design process alternatives is computationally expensive and inappropriate if the objective is to reduce the complexity of processes. Instead of executing all process alternatives, a more effective strategy is to start with the simplest design process options and progressively make it more complex based on the design needs. Hence, based on the literature, the hypothesis H1.1 is appropriate, establishing the theoretical structural validation.

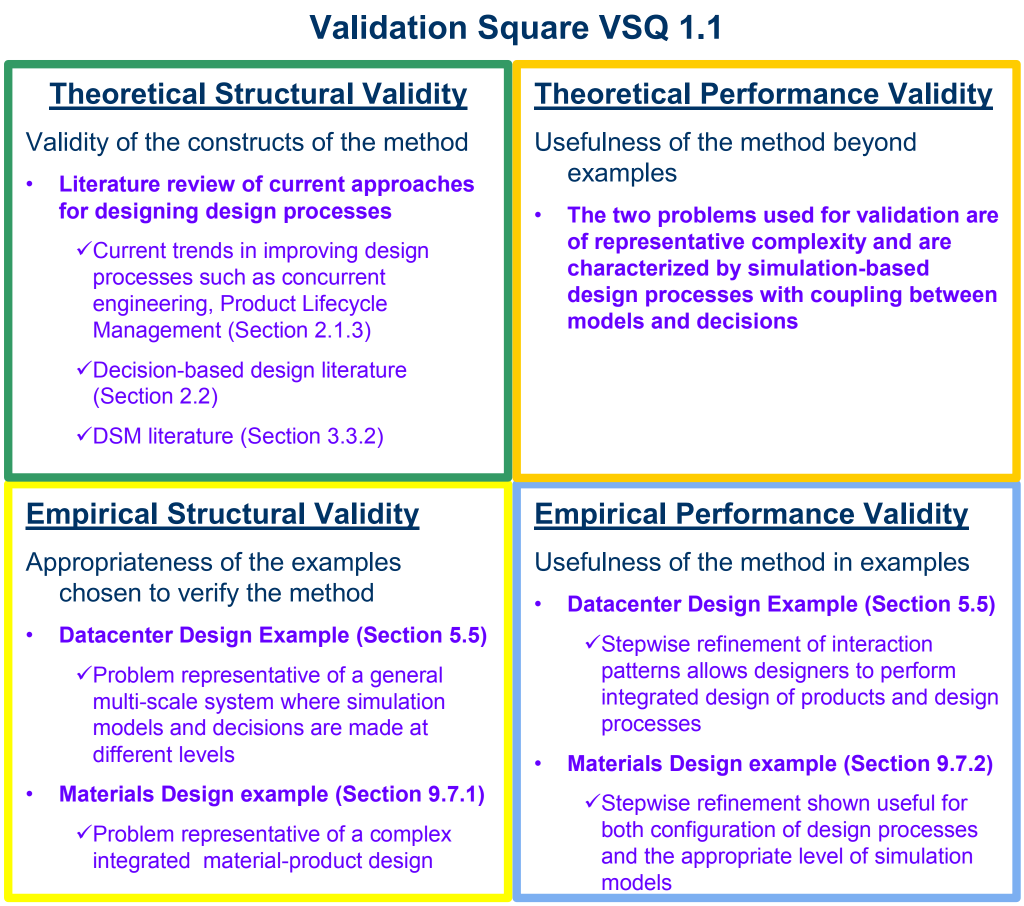


Figure 10-2 - Validation square VSQ1.1 for Hypothesis H1.1 (refer Figure 10-1)

Hypothesis H1.1 is empirically validated through application of the stepwise refinement of design processes to *datacenter* and *materials design example*. Both these problems are appropriate because they are multiscale in nature and represent different levels of complexity. The ***empirical structural validation*** of datacenter example is established in Section 5.5, and the empirical structural validation of the materials design example is established in Section 9.7.1. In the datacenter example, different types of interactions between simulation models and decisions are considered. These combinations represent the design process options at different levels of refinement. These combinations are used for designing the datacenter cooling system under different preference conditions. Similarly we explore different levels of refinement of models and decisions in the materials design problem.

Based on the results from these examples, we observe that the ***design processes that model all the interaction flows may result in similar decisions as some of the simpler design processes***. In other words, simple design processes may result in satisficing decisions. Since a stepwise refinement of the design processes is carried out, the designers don't need to execute the design process considering all interactions. The designers can get the same quality of decisions efficiently by starting with the simple design processes and gradually refining it. This implies that stepwise refinement of design processes is appropriate, thereby providing ***empirical performance validation*** of Hypothesis H1.1. Additional insight is gained from the datacenter example, where we observe that designers' decisions have a great impact on the appropriateness of design processes for a decision making scenario. Hence, error is not the only criterion for selecting design process options or for selecting appropriate simulation models.

Advantages and Limitations: The ***advantage*** of using stepwise refinement of design processes is that designers don't need to use the most complex design process for decision making. However, we may argue that more than one refinement steps may be required for determining the right level of design process. Multiple executions of design processes at increasing levels of refinement may result in total added computational costs that are greater than single execution of the most complex design process. This is a ***limitation*** of the proposed approach. Currently, the extent of refinement (in each refinement cycle) is based on the designers' experience and insight into the problem. If the refinement after each cycle is very small, then the overall computational costs (added over all the cycles) would be high, whereas if the extent of refinement after each cycle is high, the computational cost for a single process execution would be high. Hence, there is a tradeoff between the overall computational effort throughout the design and the computational effort in a single design process execution at the meta-design level. This tradeoff is not *explicitly* captured in the current framework. This limitation is addressed in a future work Section 10.4.2. It is assumed that the designers make that judgment based on the expected benefit from refinement.

Hypothesis 1.2: The second hypothesis used to answer the first research question (H1.2) is that *design processes can be designed as hierarchical systems composed of repeated building blocks defined in terms of the interaction patterns*. This hypothesis essentially consists of two parts - the first part is that design processes can be viewed as hierarchical systems composed of modular building blocks and the second part is that the building blocks can be described in terms of the types of interactions between the process blocks. This hypothesis is embodied in Step 2 of the robust multiscale design exploration presented in Section 3.5. This hypothesis is also used in the decision and scale decoupling methods presented in Sections 5.3.1 and 5.4.1. In this dissertation, nine types of interaction patterns are identified and represented in a matrix form. The columns in the matrix correspond to independent, dependent, and coupled information flows. The rows of the matrix correspond to model interactions, decision interactions, and combined model-decision interactions.

The Hypothesis H1.2 is validated using the validation square construct described in Section 1.3. The validation square (VSQ 1.2) for this hypothesis is shown in Figure 10-3. VSQ 1.2 is a validation sub-square for the overall dissertation level validation square (see Figure 10-1). Theoretical structural validation for this hypothesis is carried out by performing a literature review on Design Structrue Matrix (DSM) and patterns. Based on the literature review on DSM in Section 3.3.2, it is concluded that the design processes can be modeled hierarchically in a matix format. Using the matrix representation, the design processes can be configured based on the information flows between different tasks. Based on the review of literature on interaction patterns, it is concluded that the patterns are not yet applied to engineering design processes. By applying the concepts of DSM-based matrix information flow representation and patterns to design processes, the design processes can be modeled hierarchically in terms of building blocks and represented in a matrix form allowing reconfiguration. DSM is a construct for representing and manipulating design processes, and patterns are ways to forumulate the repeating building blocks. Both these constructs are compatible with each other because the DSM captures the inteaactions between different tasks and the patterns are defined in terms of the different types of interactions between decisions and simulation models. This ensures ***theoretical structural validation*** of the hypothesis H1.2.

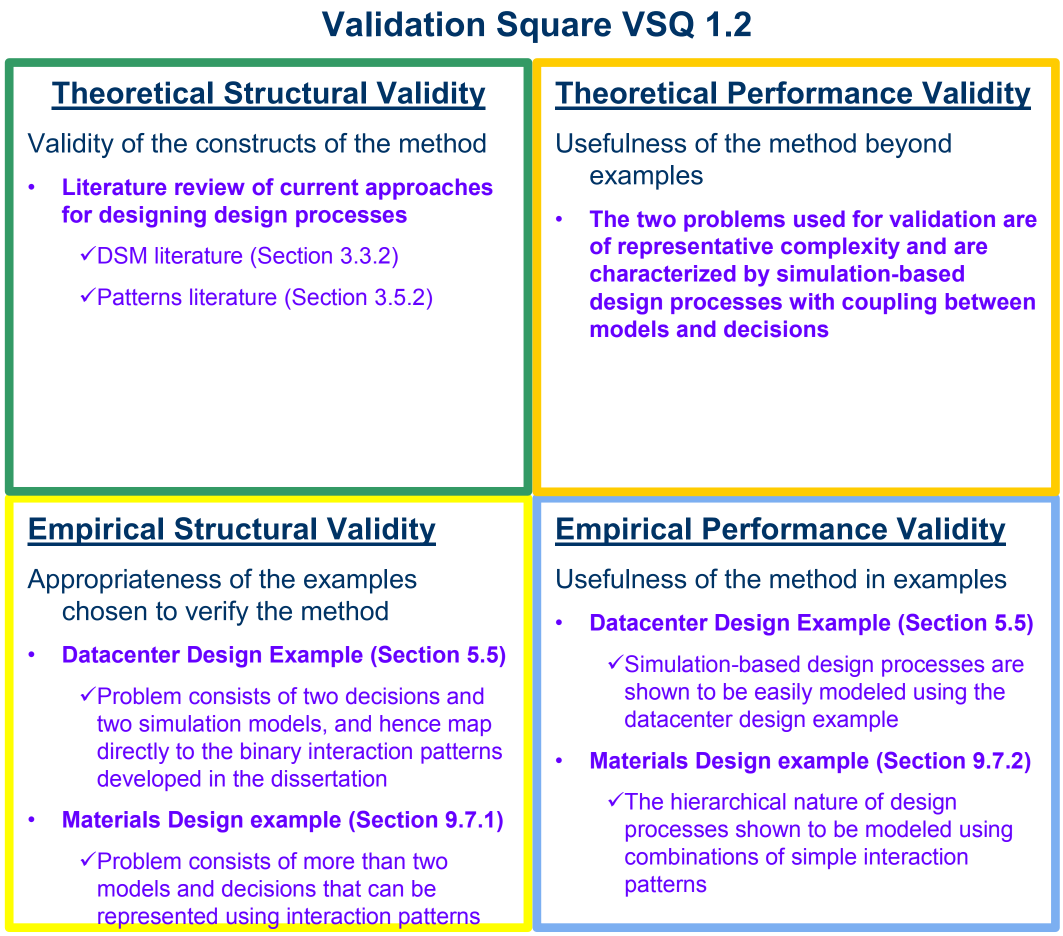


Figure 10-3 - Validation square VSQ 1.2 for validation of hypothesis H1.2 (refer Figure 10-1)

Hypothesis H1.2 is empirically validated in the dissertation using various examples. The interaction patterns are shown to be useful in hierarchically modeling complex design processes. Tools such as the Design Structure Matrix (DSM) are used for converting the graph-based representation of design processes into matrix-based representations that can be used with existing algorithms for identifying the interaction patterns between different tasks. The ***empirical structrural validation*** is discussed in Sections 5.5 and 9.7.1. The datacenter example and materials design example are chosen because of their different characteristics. The datacenter design example doesn't require hierarchical combination of interaction patterns whereas the materials design example requires hierarchical combination of interaction patterns. The datacenter design example consists of two simulation models and decisions at different scales. Hence, the binary interaction patterns can be directly applied for modeling datacenter design processes. The multiscale materials design problem consists of three simulation models, therefore requiring a combination of interaction patterns to model the design processes.

The ***empirical performance validation*** of Hypothesis H1.2 is shown by demonstrating that the matrix-based approach and the interaction patterns can be used for modeling hierarchical design processes. The interaction patterns are shown for modeling processes in datacenter example with two decisions and two simulation models supporting those decisions. The patterns are also shown for two decisions supported by three simulation models in the material-product design example (see Chapter 9). The patterns are applied to more complex design scenarios involving more than three decisions for the structure design example to show that they can be applied to other design processes. The convenience in applying the nine interaction patterns is a result of their simplicity.

Advantages and Limitations

Note that in the materials design example, a hierarchy of design process decisions - that deal with interaction patterns between simulation models and the interaction patterns between decisions - are considered. The total number of combinations of interaction patterns for the material-product design scenario is 16. Hence, there are is a variety of design process options from which the designers can select the most appropriate one. One of the ***limitations*** of the interaction patterns is that as more and more decisions and models are added, the total possible combinations may increase significantly. In other words, although the interaction patterns with interactions between two entities (models/decisions) can be used to model complex scenarios, complex patterns with more than two entities are convenient to model design processes that are more complex. These complex patterns can be developed from combinations of the nine interaction patterns presented in the dissertation.

The key ***advantage*** of interaction patterns is that they can be used to view design processes as modular systems that can be composed hierarchically in a manner similar to the assembly of subsystems to form larger systems. These interaction patterns can be used to model any simulation-based design processes that can be expressed in terms of decisions and simulation models. The ***scope of applicability*** of these constructs is simulation-based design where physics-based simulation models are available and the decisions can be formulated in mathematical form. Although the general concept of using patterns for design process building blocks remains valid for any phase in the design process, the methods and constructs such as interaction patterns are most applicable for simulation-based design. If other phases of design process are to be considered, the kinds of interaction patterns may change. Other types of interaction patterns are discussed in the future work Section 10.4.1.

10.2.2 Research Question 2 – Simplification of Design Processes and Refinement of Simulation Models

Hypothesis H2.1: The second research question is related to the metrics for simplification of design processes and the refinement of simulation models to support faster and good quality decision making. The research question is answered using two hypotheses H2.1 and H2.2. The hypotheses are embodied in the two steps of the design method – Step 3: simplify the process patterns using value of information, and Step 6:

refine models and design solution. Step 3 is further divided into specific substeps for scale-decoupling and decision-decoupling (discussed in Chapter 5) that are embodiments of H2.2. The systematic refinement and simplification is based on the value of information metric developed in Chapter 4 as an embodiment of the hypothesis H2.1. According to hypothesis (H2.1), *design processes can be simplified and models refined by making tradeoffs among value-of-information obtained via simulations and need to achieve robust, satisficing solutions*. Value of information refers to the improvement in the quality of designer's decision after addition of information via refinement of simulation model or design process. The validation strategy for Hypothesis H2.1 (VSQ 2.1) is presented in Figure 10-4. VSQ 2.1 is a validation sub-square for the overall dissertation level validation square (see Figure 10-1)

Theoretical structural validation of hypothesis H2.1 is performed by evaluating the existing literature on metrics for value of information in Section 4.2. Based on the literature review, three requirements for the value of information metric are determined. These include quantification of imprecision, consideration of deviation of payoff function in addition to the expected value, and quantification of opportunity for improving the design solution through addition of more information. In order to address these three requirements, three different components of the value of information metric are proposed in this dissertation – maximum ex-post value, achievement ratio, and opportunity ratio. The implications of different combinations of these three components of the metric are discussed in Table 4-3.

Validation Square VSQ 2.1

Theoretical Structural Validity	Theoretical Performance Validity
Validity of the constructs of the method • **Literature review of Value-of-Information metrics (Sections 4.2, 4.5.1)** ✓Value of information can be used for meta-level decisions ✓Current value of information metrics are unsuitable for simulation-based design processes ✓Requirements list for new value-of-information based metric	Usefulness of the method beyond examples • **Problems used for validation are of representative complexity and are characterized by simulation-based design processes with coupling between models and decisions**
Empirical Structural Validity	**Empirical Performance Validity**
Appropriateness of the examples • **Pressure Vessel Example (Section 4.5.2)** ✓Simple problem for demonstrating the application of value of information metric • **Datacenter Design Example (Section 5.5)** ✓Validation of metric for design process simplification • **Materials Design Example (Section 9.6)** ✓Validation of metric for simulation model refinement	Usefulness of the method in examples • **Spring Design Example (Section 4.5.2)** ✓Metric's appropriateness for meta-level decisions • **Datacenter Design Example (Section 5.5)** ✓Metric's appropriateness for design process simplification • **Materials Design Example (Section 9.6)** ✓Metric's appropriateness for simulation model refinement

Figure 10-4 – Validation Square VSQ 2.1 for Hypothesis H2.1 (refer Figure 10-1)

Empirical structural validation of these metrics is performed using various examples including the design of pressure vessel, datacenter design, and materials design. In the pressure vessel example, the value of refinement of simulation models for strength and density is calculated to guide the designers' efforts. The example is appropriate for demonstrating the applicability of value-of-information metric in meta-design decisions. The example is chosen because of its simplicity. The decision in pressure vessel design problem is based on a single design variable and hence, can be visualized using 2-D plots. In the datacenter example, the metric is used to determine the type of interaction between models and decisions. In the materials design example, the metric is used to determine the appropriate level of refinement of shock simulation model. The datacenter design and multiscale materials design examples are appropriate for validation of hypothesis H2.1 because of their hierarchical nature, which is the motivation of developing the overall framewo. Using these examples, the appropriateness of metrics for making process-level decisions is shown.

Empirical Performance Validation of the value-of-information based metric is performed by using the metric in the three example scenarios. In the pressure vessel design scenario, the metric is found to be useful in reducing the ranges of variables such as material density and strength, to a limit beyond which the impact on design decisions is negligible. It is shown that the reduction in range of density provides a consistent reduction in the value of additional information. This provides the designers with an estimate of reduction in range required. The value of information also indicates that beyond a certain reduction in range of strength, the strength does not affect the design decisions. This is a helpful insight for designers and demonstrates that the value of information can be used for making efficient design decisions.

The metric is used in the datacenter design problem to determine the appropriate levels of simplification of design process interaction patterns. It is shown that by using the metric, the designers can simplify the design processes significantly without affecting the design decisions. In the materials design scenario, the metric is used to determine the appropriate level of refinement of shock simulation model beyond which the impact on design decisions is negligible. Using the metric, we are able to determine the right level of refinement. In addition to that, using the metric, it is possible to identify that the two dimensions of refinement of the simulation model are not independent. Hence, the value of information is a useful guide for meta-level decisions.

Advantages and Limitations

The main ***advantage*** of this metric is that it quantifies the impact of process level decisions on product level decisions. Further, the metric is simple to evaluate and requires information only about the lower and upper bounds of payoff functions. There are a number of ways in which the metric can be improved. As mentioned in the previous section, one of the main ***limitations*** of the metric is that it does not quantify *how much refinement is necessary* for simulation models and design processes. The possible improvements on this metric are discussed in detail in Section 4.5.4. Note that the value of information is just one type of metric that can be used to evaluate the effect of design process decisions. Other types of metrics can be used for quantifying this effect. Such metrics are discussed in a future work section 10.4.3.

Hypothesis H2.2: The second hypothesis used to answer the second research question is that *design processes can be simplified using decoupling of scales, decisions and functionalities*. The hypothesis is embodied in the form of methods for scale and decision decoupling presented in Sections 5.3.1 and 5.4.1. The method is based on the nine interaction patterns from Hypothesis H1.2, the value of information metric developed based on hypothesis H2.1 and the general idea of starting with a simple interaction pattern and refining it if there is a possibility of improving the decision making capability. The validation square used to validate the hypothesis H2.2 (VSQ 2.2) is shown in Figure 10-5. VSQ 2.2 is a validation sub-square for the overall dissertation level validation square (see Figure 10-1). This figure is a modified version of the Figure 5-34.

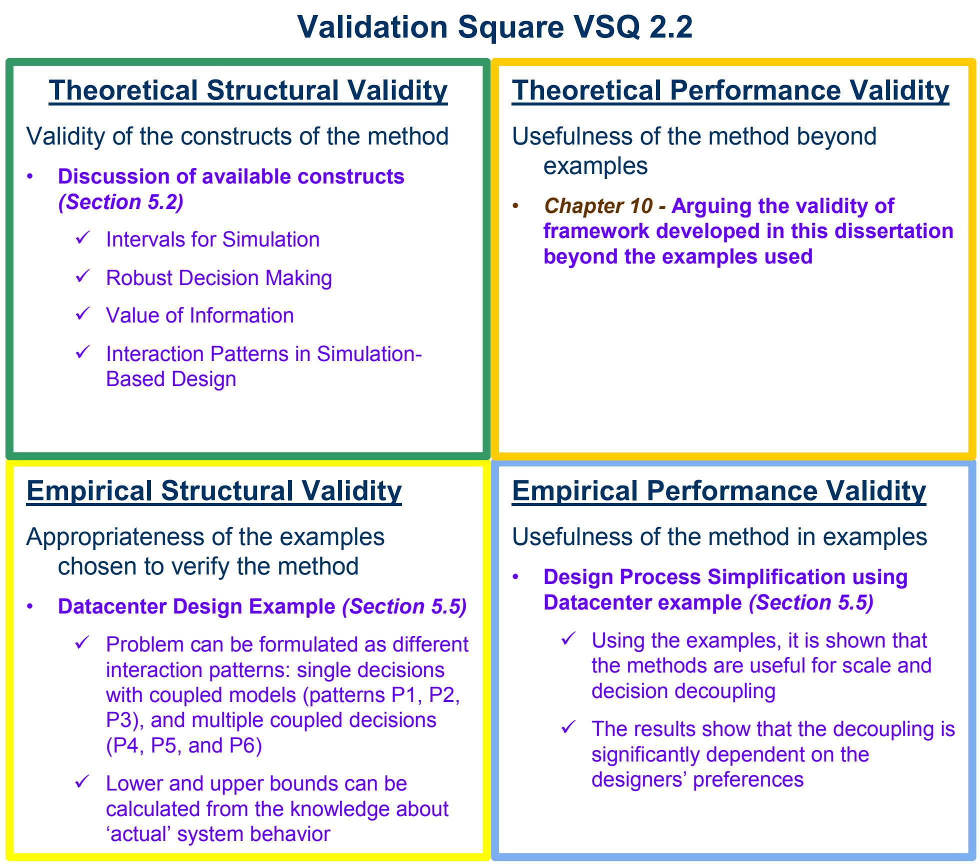


Figure 10-5 – Validation square VSQ 2.2 for hypothesis H2.2 (refer Figure 10-1)

The ***theoretical structural validation*** is performed by reviewing the existing constructs used for scale and decision decoupling. The constructs include use of intervals, robust decision making, value of information and the interaction patterns for modeling design processes. Details of the theoretical structural validation are presented in Section 5.5. The methods for scale and function decoupling are empirically validated using the datacenter design example where the information flow between two simulation models and decisions at the cabinet level and the computer level are modeled using the interaction patterns. These interaction patterns are simplified using the corresponding decoupling methods. It is observed that for different types of designers' preferences, different interaction patterns are suitable. The methods are also applied to the multiscale materials design problem involving three simulation models at different scales and two decisions related to the product and the material. The method for functional decoupling is similar to the methods for scale and decision decoupling and hence, is not considered in detail in this dissertation.

An interesting case, where functional decoupling cannot be performed due to strong coupling between the different functional behaviors, is investigated in this dissertation. In such a strongly coupling scenario, an interval-based focalization method is presented for decision making in a decentralized design scenario. The main advantages of this method include gradual reduction of design space, non divergence, reduced complexity of design process and reduced information transfers compared to its coupled counterparts.

10.2.3 Research Question 3 – Modeling and Representation of Design Information

Hypothesis H3.1: The third research question in the dissertation is related to the modeling and representation of design information in a computer interpretable format to support meta-design. The hypothesis (H3.1) used to answer this research question is that *meta-design can be supported by separating product, problem, and process specific information*. The validation strategy for this hypothesis (VSQ 3. 1) is outlined in Figure 10-6. VSQ 3.1 is a validation sub-square for the overall dissertation level validation square (see Figure 10-1).

Theoretical structural validation is performed through a literature review of design process information modeling efforts and the implementation of existing computational frameworks for simulation-based design such as FIPER, ModelCenter, and iSIGHT. The literature review is presented in Sections 2.6 and 7.1. Based on the literature review, it is *observed that one of the main reasons that make the simulation frameworks incapable of* *supporting meta-design is that the information about design products, processes, and design problems is captured in a highly integrated fashion*. Due to this coupled nature of information capture, the designers cannot utilize different design processes for solving a design problem. Utilization of different processes requires designers to completely redefine the design process. In other words, product independent representation of design processes is not supported in the current simulation-based design frameworks.

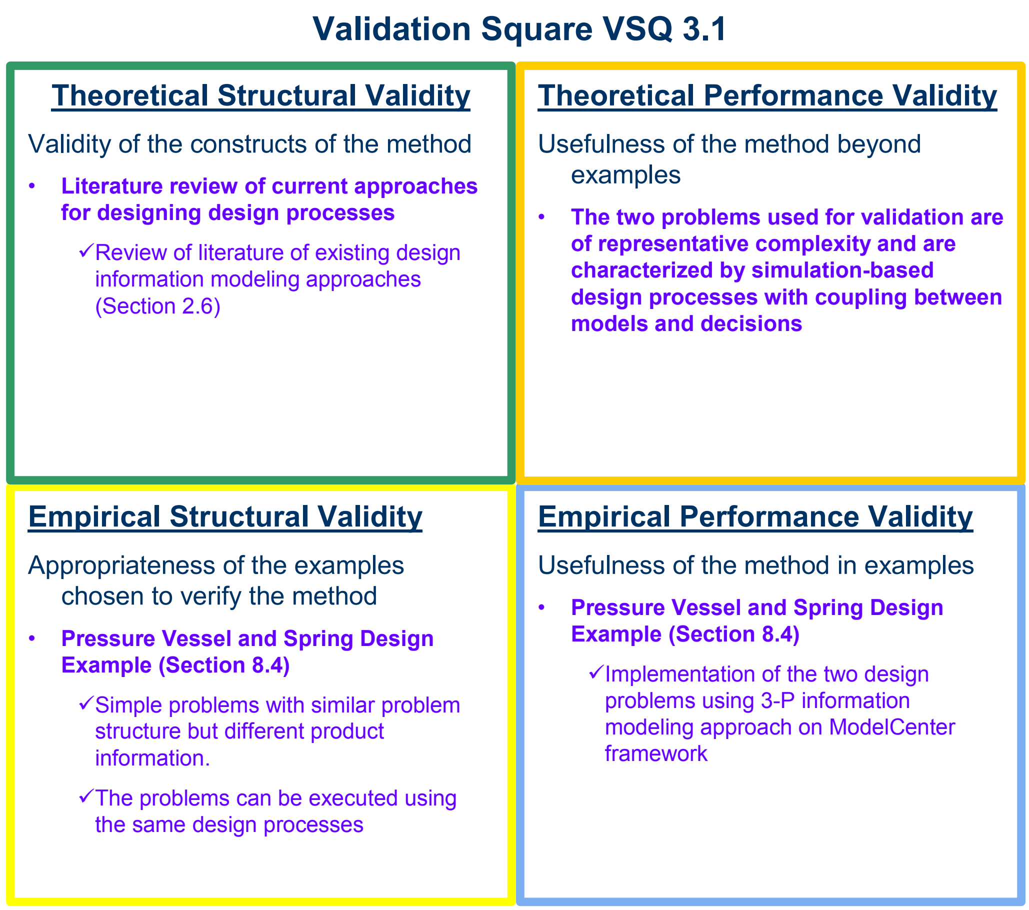


Figure 10-6 – Validation Square VSQ 3.1 for Hypothesis H3.1 (refer Figure 10-1)

This shortcoming is addressed in this dissertation by modularly capturing information about products, design problems, and design processes. Hypothesis H3.1 is embodied in the 3-P information modeling approach in this dissertation. Generic information schemas for product, problem (decisions problem only), and design processes. These information models are preliminary in nature and can be extended in the future by including more concepts and their relationships. The avenues for future research in this area are discussed in Section 10.4.6. One aspect of the approach his implemented and validated in a commercially available framework – ModelCenter. The implementation details are presented in Section 8.4. The validation involves demonstrating that a generic process representation (i.e., process template) can be used for designing multiple products. The implementation is validated for design of a spring and a pressure vessel. In that example, a single process template is developed in ModelCenter and utilized for design of these different products. Similar process templates can be developed for other design processes and used for designing a product. From the design examples, it is deduced that the hypothesis of separation of information is hepful for performing design process exploration.

10.2.4 Theoretical Performance Validation of Hypotheses H1.1, H1.2, H2.1, H2.2, H3.1

As discussed in Section 1.3, theoretical performance validation involves establishing that the proposed methods are useful beyond the example problems. This involves determining the characteristics of the example problems that make them representative of general class of problems. Based on the utility of the method for these example problems, its usefulness for general class of problems is inferred. The general characteristics of problems for the validity of constructs developed in this dissertation are:

1. The decisions can be formulated mathematically and the simulation models are available for design decision-making. This is generally true in the preliminary design phase where the concept has already been selected and mathematical models are available for predicting the system behavior.

2. Designers can express their preference in terms of utility functions. If the designers cannot express their preferences in mathematical terms, the metrics based on value of information cannot be applied.
3. Since the design methods developed in this dissertation are based on interactions between models and decisions, the methods are particularly applicable when the coupling between decisions and models is the main source of complexity in design. The methods developed in this dissertation are useful if there is a possibility of achieving time and cost savings via simplification of couplings between decisions and models. It is an assumption in this dissertation that this is generally true.
4. The simplification methods require information about the error in terms of lower and upper bounds. If this information is unavailable, the ex-post value of information cannot be calculated. Hence, this particular metric is not applicable if a number of simulation models for modeling a phenomenon are available but the information about their error is unavailable.
5. The methods developed for multifunctional design are valid for the cases where the design space is shared between different designers having conflicting objectives arising from the different functional requirements.

The example problems selected in this dissertation satisfy the characteristics described above. The framework developed in this dissertation is shown to be useful for the example problems chosen for validation. Hence, we take the leap of faith and argue that the framework is valid for other problems that satisfy these characteristics. Having discussed the validation of the hypotheses on which the answers to the research questions are based, the next step is to discuss the achievements and contributions from the research presented in this dissertation.

10.3 Achievements and Contributions

The achievements and contributions presented in this dissertation are divided into three categories – *a)* contributions to the field of design methodology, b) contributions to the field of multiscale materials and product design, and c) computer-based support for design. The contributions in the field of design methodology are related to the research questions 1 and 2 and associated hypotheses introduced in Chapter 1. These two research questions are related to the development of methods and metrics that support the integrated design of products and design processes. These methods and metrics are validated through application to various design problems. As a result of the application of these constructs, several achievements are realized in the design of specific problems. The most important problem specific achievement in the dissertation is in the domain on multiscale materials design. Finally, the third contribution is related to the Research Question 3 and associated hypothesis. These three categories of contributions are highlighted in Sections 10.3.1, 10.3.2, and 10.3.3 respectively.

10.3.1 Contributions to the Field of Design Methodology

The primary research contribution corresponds to the principal goal, primary research question, and primary research hypothesis – **a framework for integrated design of multiscale products and design processes, to facilitate the effective utilization of information and computational resources in preliminary design of multiscale systems with potential applications to other complex systems.** The primary contribution is explained in greater detail by expanding it into several secondary research contributions in the field of design methodology. A summary of the contributions from this dissertation in the field of design methodology are categorized into strategies, methods, and metrics. The contributions are listed in Table 10-1.

Table 10-1 – Summary of contributions from the dissertation

Contribution
Design Methodology
Design method is created for simulation-based, integrated design of products and design processes *(Section 3.5)*
Explicit accounting of metadesign decisions, meta-design in the context of designers' preferences *(Section 4.3)*
Extension of robust design applications to collaborative decision making *(Section 5.2.2)*
New value-of-information based metrics *(Section 4.3)*
Methods for using value of information based metrics for making design process related decisions *(Sections 5.3.1, and 5.4.1)*
A set-based focalization method for multifunctional design *(Section 6.4)*
Multiscale Materials and Product Design
Systematic method for design of multiscale materials with consideration of products and design processes *(Section 3.5)*
Formalization of a design process for the multiscale materials *(Section 9.5)*
Simulation models at different scales integrated together; connections exercised for a preliminary design of the materials and products *(Section 9.4)*
Development of projectile level simulation model *(Section 9.4.3)*
Strategy for designing materials-products-design processes *(Section 9.5)*
Particle level shock simulation model subjected to a detailed refinement study *(Section 9.6)*
Basic structure for the design of multifunctional energetic structural materials *(Section 9.5)*
Computer-Based Support for Design
3-P information modeling strategy *(Section 7.3)*
Separation of declarative information and procedural information applied to design processes *(Section 7.2.3)*
Augmentation of currently available commercial tools *(Section 8.4)*

A comprehensive ***design method is created for simulation-based, integrated design of products and design processes***. The method is based on four foundational elements – designing design processes in association with products, applying systems-based approach to meta-design, reliance on robust solutions as opposed to optimum solutions, and the systematic refinement of design. The primary strength of this method is the systematic consideration of design-process related decisions during the design of products. Such meta-level decisions are made by assessing the performance of processes in terms of metrics. Meta-design is carried out based on the assumption that design processes themselves are hierarchical systems that can be modeled in terms of standardized process patterns. One such class of process patterns used in this dissertation is based on interaction types. Standardized building blocks of a design process are identified in terms of interaction patterns between process components. Various interaction patterns between simulation models and decisions are identified and used to model design processes for different design problems. These interaction patterns facilitate modeling design processes, evaluation of the impact of couplings between different components of design processes and allow process related decisions such model decoupling, decision decoupling, etc.

The important differentiating features and advantages of this method is that it enables designers to *a)* ***explicitly account for metadesign decisions*** during a design process, and *b)* allow ***meta-design in the context of designers' preferences***. Hence, the design processes are designed specifically for the product design under consideration. The method allows designers to consider multiple objectives in meta-design. This is an enhancement over existing design methods that either do not consider the design of design processes at all or only consider the configuration of processes only based on streamlining information flow.

The method ***implements and extends the application of robust design***. In the proposed method, design decisions are made by considering not only the average system performance, but also the variation in performance due to various sources of uncertainty. These sources of uncertainty include variability in the external noise variables, variability in the design variables, variability in the simulation models or due to propagation of uncertainty along the design chains. Robust design for handling this type of uncertainty exists in the literature and is incorporated in the design method developed in this dissertation. These types of robust design are labeled as Type I-IV robust design. In addition to these four types of robust design, an additional type of robustness is addressed in this dissertation – robustness to decisions made by other designers' decisions. ***By making decisions robust against the uncertainty in other designers' decisions, the concept of robustness is extended in this dissertation to collaborative decision making***. This type of robustness is useful when coupled decisions are decoupled during simplification of design processes. This reduces the complexity of design processes and increases the efficiency with which resources are used.

As discussed earlier in this section, the performance of design processes is measured in terms of metrics. One such metric investigated in detail in this dissertation is the value of information that is used to quantify the impact of additional information on designers' decision making capability. ***New value-of-information based metrics*** are developed for making meta-level decisions. These metrics quantify the maximum possible improvement in final design decision in terms of overall payoff. Utility theory is recommended as a means for measuring designers' overall payoff. The metrics also quantify the level of achievement of the design goals in the presence of uncertainty. These metrics allow comparison of available process options when modeled as subsequent information addition scenarios. Different process options are evaluated against each other by modeling the simplest design process as the available information and all other process options as 'additions of information' available for decision making. The effect of this additional information on decisions is calculated and used as a criterion to decide whether a process option is appropriate or not.

The process decisions considered in detail in this dissertation include *a)* decoupling of interaction patterns (both simulation models and decisions) and *b)* refinement of simulation models. In order to support these design process related decisions, ***methods for using value of information based metrics for making*** are developed in this dissertation. These methods are used to simplify design processes to a level that reduces the complexity of design processes without affecting the quality of decisions. The key strategy in these methods is to perform a tradeoff analysis between the possibility of improving the design solution and the increase in cost of improving the design process. The application of these methods require characterization of the information generated by simulation models in terms of lower and upper bounds within which the response lies. The methods for decoupling of interaction patterns and refinement of simulation models are used as subprocesses in the overall method for integrated design of multiscale products and design processes.

The decision decoupling methods are also applicable to functional decoupling in a multi-functional scenario. However, in the cases where the system is strongly coupled and the decisions cannot be decoupled without affecting the final design, the methods for designing design processes developed in this dissertation are not valuable. Such strongly coupled cases are addressed in this dissertation via ***development of a set-based focalization method for multifunctional design***. The method is based on systematic reduction of the design space sequentially by different functional experts until the design converges to a single point. The method is developed to overcome limitations of existing point-based iterative methods and other game-theoretic methods for multifunctional design. The advantages of this method include independence of convergence characteristics from initial design, and non-divergence. Further, since the method involves a gradual reduction of design freedom, it supports limited changes in the design requirements along the design process. The method builds on the concepts of interval arithmetic and box-consistency.

10.3.2 Contributions to the Field of Multiscale Materials and Product Design

As a result of applying the design methods to materials design problem, contributions are made to the field of materials design and to the general multiscale design problem for energetic structural application. The contribution specific to the field materials design is a ***systematic method for design of multiscale materials with consideration of design processes***. In this dissertation, the design of multifunctional materials is also carried out in association with the product (the projectile). This concurrent design of products, materials, and design processes is a contribution to the field of materials design. The design is carried out by considering robustness to uncertainty from various sources including error in simulation models, variability in simulation models, and uncertainty in decisions made by designers from other functional domains.

Apart from these general contributions in the field of materials design, various contributions specific to the multiscale, multifunctional materials design problem have resulted from this dissertation. The research presented in this dissertation is instrumental in the ***formalization of a design process for the multiscale materials***. The information flows between various simulation models are identified. Three ***different simulation models at different scales are integrated together*** to make decisions about the material and system level design variables. Two models – a particle level shock simulation model and the non-equilibrium mixture theory model are developed by other researchers. The third model – ***projectile level simulation is developed*** as a part of this dissertation. This simulation model can be refined and used in future design explorations.

Connections between various models are developed at the software and conceptual level (design process level) in association with Ryan Austin (Austin 2005), Haejin Choi (Choi 2005) and Jim Shepherd. In this dissertation, these ***connections are exercised for a preliminary design of the materials and products*** by considering a subset of design variables. The results of the preliminary design of materials and products are presented in this dissertation. This information will be used in the future to perform a detailed design of the material-product systems. The knowledge about the ***strategy for designing materials-products-design processes*** is useful for any multiscale design problem. Finally, the ***particle level shock simulation model is subjected to a detailed refinement study*** in the context of a design sub-problem. The strategy adopted for refinement of this simulation model can be applied to other simulation models at different scales. This is useful in the materials design project because there are various models at different stages of development. The refinement of these models requires either development of models at smaller scales or generation of experimental data. Amount of effort for refinement can either be reduced or targeted if the approach is applied to determine the critical models that should be refined and no-so-critical models that do not need to be refined. This is true in any multiscale design scenario, and hence is also a contribution to the general multiscale design domain. As a summary, the research presented in this dissertation provides ***basic structure for the design of multifunctional energetic structural materials*** and can be extended to satisfy the complete requirements list for the design.

In addition to the materials design validation example, various other examples are used throughout the dissertation to either demonstrate or validate the use of methods developed. These examples include multiscale datacenter cooling system design, pressure vessel design, linear cellular alloy design and structure design. The results from application of design methods to these examples can be utilized in the future design of corresponding products. For example, in the datacenter example, the consideration of products and design processes simultaneously for decision making is an achievement for this dissertation. It is shown that for different designer preferences, the required levels of complexity of design processes are different. Hence, it is shown that it is not required to consider the most detailed simulation models. The modeling and computational effort at the smallest scales can be reduced in a variety of design scenarios. We believe that this is an important result for the domain of datacenter design.

10.3.3 Contributions to the Field of Computer-based Support for Design

In order to support meta-design in the computational frameworks, we present a ***3-P information modeling strategy*** in this dissertation. It involves separating the information about the product, design process, and the design problem. The general mindset for developing the information modeling strategy is that design is a network of transformations that transform the product information from one state to another. Various kinds of transformations are identified in this dissertation that include decisions, abstraction, decomposition, etc. Each of these transformations is associated with a design problem. The general idea of problems is adopted from the Decision Support Problem Technique literature. The transformations are associated with processes for execution. Only one design transformation is discussed in detail in this dissertation – the decisions. Preliminary information models are developed for products, decision problems, and processes. By separating the information about problems and processes, we essentially ***separate the declarative information from the procedural information***. The *3-P* modeling approach, proposed in this dissertation, enables designers to capture design process information in a manner that allows quick process reconfiguration, thereby supporting design process exploration. The modular separation of information associated with problem, product, and processes enables exploring different design sub-processes for solving a given design problem. The key advantages of the *3-P* approach arise from the three basic ideas used for its development (extension of DSP Technique, modular template based approach, and separation of declarative and procedural information). These advantages include the following: a) information related to Problems, Products and Processes is separated and captured via modular templates, b) different combinations of Problem, Product, and Process declarations can be combined together to generate specific computationally executable processes, c) process knowledge can be captured and reused across problems and products, and d) the information model allows composability of instantiated sub-processes into higher level processes.

In addition to the independent use of the proposed approach, it can serve as an ***augmentation of currently available commercial tools*** such as iSIGHT (2004), FIPER (Engenious Inc. 2004) and Model Center (Phoenix Integration Inc. 2004). We emphasize here that only one aspect of the 3-P modeling strategy is validated in this dissertation through implementation in ModelCenter. The aspect involves showing the usefulness of separation of declarative (problem specific information) and procedural (process specific information). By separating the problem and process related information, it is shown that the same processes modeled in ModelCenter can be used for many different design problems. Other aspects of the approach such as implementation of the product information models and need to be validated and are discussed in the future work section 10.4.6.

10.4 Limitations and Opportunities for Future Work

Although a diverse set of topics ranging from design methods to information modeling strategies (all topics emerging from a common thread of integrated design of products and design processes) are addressed in this dissertation, we believe that this dissertation is only a small step towards the fulfillment of the vision.

Table 10-2 – Opportunities for future work and expected time frame

Section No.	Opportunities for Future Work	Expected Time Frame
10.4.1	Extending the Design Process Building Blocks	2-5 Years
10.4.2	Enhancements to the Value-of-Information Metrics	0-2 Years
10.4.3	Designing Open Design Processes	2-5 Years
10.4.4	Extending the Method for Interval-based Focalization	0-2 Years
10.4.5	Simulating Design Processes	5+ Years
10.4.6	Implementation of 3-P Approach in a Software Framework	0-2 Years
10.4.7	Synthesis of Design Processes from Existing Elements	2-5 Years
10.4.8	Extending the Same Process Design Methods to Other Types of Processes	2-5 Years
10.4.9	Different Types of Characterization of Models	0-2 Years
10.4.10	Designing Families of Design Processes	2-5 Years
10.4.11	Organizational Impact of Designing Design Processes	2-5 Years

There are many limitations to the breadth and extent of the present body of work. These limitations offer a host of opportunities for future work. Some of the possible directions for future work are outlined in Table 10-2 and discussed throughout this section. In the table, time-frame for each research activity is also listed.

10.4.1 Extending the Design Process Building Blocks

The general concept followed in this research is that design processes can be viewed as modular, hierarchical systems. The characteristic of any system is that it has subsystems and interfaces between these subsystems. In this dissertation, we have defined the reusable design process building blocks in terms of different types of processes elements (simulation models and decisions) and different types of interfaces (i.e., information flows that can be independent, sequential, and coupled). The building blocks of the design process are called interaction patterns. Interaction patterns are chosen as building blocks in this dissertation because of their importance in the multiscale design. Interaction patterns are only one way in which these building blocks can be defined. The questions that arise are – a) apart from models and decisions, what are the other types of process blocks (activities) that can be used to model design processes?, and b) are there other types of interfaces between these? The answers to these questions result in other types of building blocks that can be used for designing processes.

Notice that in this dissertation, our focus is mainly on simulation-based design processes. If the ***scope is extended to general design processes***, the building blocks may include abstraction, problem decomposition, synthesis, ideation, etc. These general information transformations of design processes are briefly discussed in Section 8.2. All these transformations have a potential to become design process building blocks. If the scope is further extended to ***general processes***, many more process building blocks can be identified. The building blocks are also dependent on the types of processes modeled. For example, the same idea can be applied to manufacturing processes where specific manufacturing activities can serve as building blocks. In the domain of supply chains, the building blocks can be defined in terms of activities such as order management, inventory management, marketing information management, sales and support, etc. The efforts to define and standardize building blocks for supply chain processes are already underway as a part of the Rosetta Net standards called Partner Interface Processes®.

In addition to different building blocks in different domains and types of processes, the interaction patterns can also be defined at different ***levels of abstraction***. In this dissertation, we define general the building blocks using simulation models and decisions in general. These building blocks can be made more specific by assigning specific simulation models and specific decisions. For example, for the materials design domain, the building blocks can be defined as interaction between micro-scale and the nano-scale models. At this level of abstraction of interaction patterns, it is possible to assign specific parameters as information flows between models at different scales. Hence, we believe that there is a need for modeling interaction patterns at different levels of abstraction. By defining the interaction patterns at different levels, designers can benefit by reusing domain specific information available in the design process building blocks. Further, new process building blocks can be obtained by ***composing existing building blocks*** into more complex building blocks. These composite building blocks can be used to design processes in a hierarchical fashion.

10.4.2 Enhancements to the Value-of-Information Metrics

The value of information metrics used in this dissertation can be improved further to be used in more complex design scenarios. The metrics used in this dissertation are based on the lower and upper bounds on values that can possibly be achieved. Hence, the resultant metric is very conservative. Further, in the cases where there are ***multiple***

sources of imprecision in the simulation models, the metric provides information about the overall improvement only. It does not support quantification of the influence of individual dimensions of refinement. Hence, based on the metric, the designers cannot directly determine *how* the model should be refined. It only provides information about *whether* the model needs to be refined or not. The ***cost*** of improving a simulation model and the ***time savings*** through simplification are not included in the value of information metric in this dissertation. The current value of information metric only includes the information about benefit achieved in terms of the overall utility. However, this does not require a change in the way this metric is applied. The only change that is required is to include cost and utility functions in the calculation of the overall utility.

The reliance on lower and upper bounds for the use of metric currently restricts its use for cases where information about these bounds is unavailable. Hence, some scenarios where this information is unavailable are not handled using the metric. For example, consider a scenario we have ***multiple simulation models that embody different mathematical models*** for the same physical phenomena, predicting the same response value. All the models have some level of imprecision but the most accurate model is unknown. Hence, for each model, the information about the upper and lower bounds of error is not available. In this case, the current value of information needs to be extended. One possibility of improvement of the metric (reducing the conservative nature) is by taking into account the possibility of achieving different values between the lower and upper bound. ***Designers' insight*** in this regard can be quantified using ***distributions of belief*** over the model's output (belief functions). We believe that this idea can be implemented using Dempster-Shafer's theory.

10.4.3 Designing Open Design Processes

In this dissertation, the design of design processes is based only on a single metric – value of information. However, there can be various other ways in which the effectiveness of design processes can be measured. One such class of metrics can be derived from the design of open engineering systems. Open engineering systems are systems of industrial products, services and/processes that are readily adaptable to changes in their environment which enable producers to remain competitive in a global marketplace through continuous improvement and indefinite growth of an existing technology base (Simpson 1998). Note that both products and the design processes can be designed as open systems. The idea of open engineering systems is applied to design of products but its application for design of processes is not available in the literature. The key to designing open engineering systems is *adaptability to changes in the environment*. The environment for a product is a set of conditions in which it is being used. Hence, a product is open if it is adaptable to changes in the conditions in which it is used. The environment for a design process includes the product which is being designed, the considerations used to design a product (like robustness, reliability, etc). This implies that if a similar product is being designed or the same product is being designed with added considerations, the process should not change. Hence, a process is open if it can be used to design similar products and same products with different design considerations.

Various techniques like robustness, modularity, maintaining design freedom, adaptability, etc. are proposed for achieving openness in a system (Simpson 1998). Hence, openness of systems can be measured by developing quantitative metrics for these. The quantitative measures related to openness of a product are: *design freedom*, *robustness*, *complexity*, *modularity* (which is closely linked to complexity) and *coupling*.

Some of the metrics for measuring these quantities available in the literature are discussed next.

1. *Design Freedom*: Design is the process of a series of modifications (expansions and reductions) in the design space to achieve a goal. The design space determines the freedom a designer has to modify the product. In other words, design freedom determines how open the product is to changes in the environment, specifications etc. Design freedom is defined as the extent to which a system can be adjusted while still meeting its design requirements (Simpson, Rosen et al. 1998). Simpson and co authors (Simpson, Rosen et al. 1998) define the design freedom as the overlap between the performance of feasible designs and the range of initial targets.

   Wood in (Wood 2000) defines design freedom as a measure of the size of the design space implied by the design specification. Wood argues that design freedom is a very valuable asset in design and should be managed efficiently. In another paper (Wood 2001), Wood has shown how different design methodologies are inherently linked to design freedom and that there is a need for systematically accounting for design freedom in the design process.

   Efficiency and effectiveness of a design process can be improved by delaying design commitment (Ward, Liker et al. 1995). This means that it is desirable to configure a design process such that the design freedom is maintained further along in the design process while still increasing knowledge. These ideas are embodied in the set based design philosophy where a set of specifications, design parameters and design variables are considered and narrowed down systematically (Ward, Liker et al. 1995; Liker, Sobek et al. 1996).

The importance of design freedom in designing open engineering systems is clear. It is also clear that the design process has a great impact on design freedom. In this research, we will investigate existing metrics for measuring design freedom and apply them to the various design transformations described in Section 1. The capabilities of these metrics for quantifying design freedom along a design process will be investigated and new techniques will be developed if the existing ones are not suitable for analyzing design processes.

2. *Robustness*: The Second Toyota Paradox (Ward, Liker et al. 1995) shows that passing ranged sets of specifications in a design process is more robust than passing point specifications. The process is relatively insensitive to uncertainties and requirement changes along time, which means that the process is robust. Taking a closer look at the two scenarios, we can see that the difference between these two kinds of design processes is the manner in which design freedom is managed along the process. A process in which the design freedom reduces abruptly is less robust as compared to the process where a range of design is passed. Hence, robustness of a design process is linked to the design freedom. Currently, there are no metrics for measuring such robustness of design processes. We need design freedom-based metrics for measuring the robustness of design processes.

3. *Complexity, modularity and coupling*: Complexity of a system is directly linked with the number of interconnected and interwoven parts (Rechtin and Maier 1997), (Simon 1996). Similarly, a process complexity can be defined as the amount of interactions between various activities. In a design process, the interactions between activities are through the design space. The more the design space is shared between activities, the

more is the complexity. So, it is our feeling that complexity of a design process is also linked with the design freedom and the manner in which design freedom is shared between designers.

Complexity is also related to modularity. Various researchers have developed modularity metrics for products. Newcomb and co-authors (Newcomb, Bras et al. 1996) have developed measures for modularity of products from various viewpoints (like assembly, service, recycling, etc.) and combined them together into an overall modularity metric. As mentioned, complexity is also related to the coupling between components. Understanding coupling is crucial for developing architectures robust to future changes in customer requirements (Martin and Ishii 2000). Martin (Martin and Ishii 1997; Martin and Ishii 2000) developed metrics for measuring coupling between components of a product. Other research efforts towards developing metrics for complexity include (Elmaghraby 1995) and Maimon and Braha (Braha and Maimon 1998). In this research, we will use existing metrics for measuring complexity and develop new metrics if needed.

From the review of literature on quantitative measures of openness, we find that previous research efforts are mainly focused on quantifying the openness of products but *openness of design processes has not been addressed in literature*. This leads to the following research question: "How can the openness of the products and associated design processes be quantified?" We believe that since both products and processes can be viewed as systems, existing metrics can be used directly or modified appropriately to measure openness of design processes.

10.4.4 Extending the Method for Interval-based Focalization

The set based-focalization method for multifunctional design developed in this dissertation provides various opportunities for further work. The method presented in this dissertation is only limited to cases where designers have single Nash equilibrium. Hence, one of the possibilities is to explore cases with ***multiple Nash equilibria***. A strategy for implementing cases with multiple Nash equilibria is discussed in Section 6.5.4. The strategy involves splitting the design space into subspaces when the box-consistency does not allow elimination of parts of the design space. The idea of splitting the design space can also be applied to general scenarios involving single Nash equilibrium. Design space splitting would allow concurrent exploration of design space as opposed to the sequential and cyclic exploration of design space as presented in the dissertation. In other words, the elimination step in the method presented here should be augmented with elimination and splitting of parts of design space.

An evaluation of the convergence of the focalization method as opposed to the point based methods is presented in the dissertation. The evaluation primarily indicates whether the solution will converge or not. A detailed ***study of the rate of convergence*** would be helpful to determine how much time would be required for the focalization method to converge. The convergence study at each cycle would provide insight into the rate of convergence at each cycle. This is important because for non linear problems, the rate of convergence can be different at different points in the design space. This would provide insight into the need for ***dynamic shift of control*** as the problem characteristics change by reducing the design space. This is in contrast to the static control assignment considered in this dissertation. In the current method, the design space is reduced only by using the box consistency principle. Hence, only those parts of the design space are eliminated that do not satisfy a designer's requirement. This approach is suitable for scenarios where this elimination reduces the design space significantly. However, in cases where only a small portion of the design space is reduced, other metrics for elimination may be required. For example, the elimination criterion may be set to removing the portions of the design space that satisfy a designer's threshold criterion. The threshold criterion may be updated (made more stringent) after each cycle.

In Section 6.5.4, we mention that the method is helpful in scenarios where the requirements change with time. This is because at a given time along the design process, the method allows keeping the design freedom open (as against the point-based method where only a single point in the design space is considered). A detailed ***investigation on changing requirements*** and its impact on the method is required. Since the method is based on interval arithmetic, which is inherently computationally expensive, it is important to consider the efficiency related issues in the method. Further, scenarios with more than two designers and designers with shared variables are interesting extensions to the proposed method.

10.4.5 Simulating Design Processes

In this dissertation, the design of design processes is carried out based on the indirect metrics such as value of information. This metric is used to make the tradeoff between the cost of adding more information and the possibility of achieving benefits in terms of the better decisions. The design of design process would be more efficient if instead of using the indirect metrics, the design processes can be directly simulated for some direct metrics such as cost and time. This approach of simulating design processes would be similar to simulating the behavior of products using physics-based models. Models for design processes similar to the physics based models for products are unavailable. The idea of simulating design processes is similar to the simulation of discrete event processes. In discrete event processes, the information about occurrence of events is available in statistical form. Using this statistical information and applying methods such as Markov chains, the processes can be analyzed to determine the probability distribution of final events. Such a method for simulating design processes is non-existent in the literature. An example application of simulation of design processes is in distributed multifunctional design processes. In such decentralized design scenarios, the design processes can be simulated for determining whether the process will converge or not. The convergence of decentralized design processes can be studied using control theory. We acknowledge that the simulation of design processes is a complex task and requires the development of mathematics to represent design process related information and its transformation. The research on design equation is a step towards that direction.

10.4.6 Implementation of 3-P Approach in a Software Framework

The implementation of the 3-P approach in this dissertation is limited to the reuse of processes for different products. The reuse of product and problem related information for different processes is not shown. Further, only a single decision is considered for demonstrating the application of processes for different products. The future work involves showing the processes involving multiple decentralized decisions in a design process, interacting through protocols such as game theory.

Our vision is to develop a software framework that embodies the components of the framework for integrated design of products and design processes including the methods, metrics, and information modeling approaches. The framework should be able to automatically configure the design processes based on the design problem specified. The framework should be open so that new metrics and methods can be implemented in the framework. Further, the framework should be based on the distributed collaborative design processes where design process activities are carried out by distributed agents. Constructs for configuring design processes in a distributed environment would support configuration of information flow between agents, determine their precedence and select appropriate agents from the available list of agents.

10.4.7 Synthesis of Design Processes from Existing Elements

Similar to designing products, the design of design processes involves three phases – analysis, synthesis, and evaluation. This paradigm for design processes is shown in Figure 1-8. The synthesis of design processes is used in this dissertation synonymously with generation of alternatives. The alternatives generated in this dissertation are only based on design process building blocks. It is assumed that the designers know what the flow of information is, and can easily generate the process alternatives. This is generally true when designers have some experience with the problem under consideration.

If there is a completely new design scenario, such information about the alternatives may not be available. In such scenarios, the primary task is to generate the process alternatives. We believe that this is one of the most important future challenges in the design of design processes and needs to be investigated further. The task of designing design processes in general is broader than just selecting the design process alternatives from the available options.

10.4.8 Extending the Same Process Design Methods to Other Types of Processes

The focus in this dissertation is entirely on designing products and design processes. However, in a product lifecycle, there are various kinds of processes such as the supply chain processes, manufacturing processes, procurement processes, maintenance, customer relationship processes, etc. All these processes can be designed in a manner similar to the design processes. Significant literature is already available in the field of modeling manufacturing processes as discrete event systems and optimizing them using statistical information. Efforts for modeling supply chain processes are underway by the RosettaNet community[5]. In these efforts, the objective is to develop *a)* modular building blocks for supply chain processes in a manner similar to interaction patterns defined in this dissertation, and *b)* metrics for evaluation of collaboration between supply chain partners. Future research opportunities lie in developing systematic frameworks for designing other types of processes and in integrating the design frameworks developed separately into a consistent "product lifecycle design" framework.

10.4.9 Different Types of Characterization of Models

In its current form, the simulation models are characterized using the lower and upper bounds of information. This information is used to determine the upper bound on value of information. However, in many cases, the information about lower and upper bounds are not available. In various circumstances, the designers have access to different kinds of models that embody different assumptions but the error bounds on those models are unavailable. In such cases, the value of information metric developed in this dissertation cannot be used directly. In other cases, higher fidelity of information about the output of

---

[5] www.rosettanet.org

simulation models may be available. For example, if the probability at which the output assumes certain numerical values is available, then a probabilistic estimate of improvement in decision with the improvement in the model can be calculated, which is a better estimate of the value of information.

Further investigation also needs to be carried out for different types of refinement of simulation models. Each model can be refined in various directions: by considering additional physical phenomena, adding more parameters, coupling physical phenomena, developing better models of physical phenomena, etc. The characterization of models should be such that the designers should be able to isolate the effect of different types of refinement. This is important in order to make decisions such as "along which dimension should the simulation model be refined?" Currently, the combined effect of refinement is calculated. If different types of characterization of information are available, better metrics for design processes are possible.

10.4.10 Designing Families of Design Processes

The objectives for designing design processes considered in this dissertation are to design customized design processes for specific design problems. It is shown that the design of design processes depends significantly on the preferences of designers for different objectives. Hence, by changing some design requirements, the design process may change significantly. This process alternative would be the best alternative for an organization designing a single product. However, this may not be the most effective way of designing for an organization designing more than one product with different sets of requirements. In such a case, the designers either need to design a single design process that would satisfy on an average, the requirements for all products, or, the designers can design a family of design processes that can be customized easily based on the requirements. The idea of product families has been recently exploited in design research but the design of ***design process families*** has not been exploited and is an exciting research opportunity.

10.4.11 Organizational Impact of Designing Design Processes

The design of design processes is significantly interlinked with the design of organizational structure. This is because the flow of information between different experts in the organization is restricted by the organizational boundaries. The design of design processes is considered in this dissertation entirely independently of the organizational structure. It is assumed that all the interactions impose equal amount of complexity in the design processes and hence, are treated equally. However, depending on the manner in which organizational interaction is set up, some interactions may introduce more complexity in the design processes than others. Hence, ideally the design of design processes and the organizations should be carried out in a parallel fashion. This may not be possible in all cases. In some cases, the design of design processes dictates the organizational structure. In such cases, the architecture of design processes serves as a guide based on which, the design processes must be structured. In other scenarios, the organizational structure dictates the design of design processes. In those cases, the organizational structure imposes additional constraints on the design of design processes.

10.5 A Vision for Research in Engineering Design

Over the past few decades, research in engineering design has progressed from consideration of single objectives to the facilitation of tradeoffs among multiple conflicting measures of merit. In the realm of engineering design, efforts currently center on product perfection. What is the next frontier? There are several possibilities, ranging from the global (and concurrent) design of multi-functional, multi-scale systems, whose performance is governed by phenomena spanning the various domains of their constitution, to series of dependent and interdependent events determining the performance of the products in which they culminate. While there are countless possibilities of concentrating research efforts, each worthy in their own right, we focus on the optimization of *Product Development Processes* (PDPs). Process improvement has thus far been limited to domain specific applications (e.g., manufacturing, logistics, supply chains, etc.). A central aim in engineering optimization is that of improving resource utilization in the achievement of specific objectives. We assert that the majority of resources are committed during a product's design. With this in mind, we believe that the next logical phase in the evolution of engineering design should be focused on the consideration of design processes in addition to products.

10.5.1 Design Processes as an Enterprise's Primary Intellectual Capital

A fundamental prerequisite for the sustained improvement of the resulting product-process systems is the ability to leverage the intellectual capital constituting the required level of system understanding. During the last decade, a strategic business approach for the effective management and use of corporate intellectual capital has emerged. This approach has come to be known as Product Lifecycle Management (PLM) and promises to further a holistic consideration of product design, emphasizing integration, interoperability, and sustainability throughout a product's lifecycle in order for an engineering enterprise to remain agile with respect to the constantly evolving demands of a global market. ***Intellectual capital thus far has been comprised mainly of product***

related knowledge and exploited mostly via the reusability and scalability of existing products through product platform and product family design. However, we strongly believe that focusing solely on product knowledge is not sufficient and limits agility to variant design (and adaptive design, to a limited extent). In order to effectively support the generation of entire portfolios of products (via derivative and original design), ***we believe that the design process should also be considered to constitute a crucial component of an engineering enterprise's intellectual capital***. Hence, we propose a paradigm shift that is centered on leveraging design process knowledge derived from previously designed products for entirely new products, thereby greatly reducing the computational burden associated with design.

The sustained improvement of Product Development Processes (PDPs) has long been the focus of manufacturing and more recently that of design as well. This is due in part to the key realization that a PDP constitutes not only a central component of the engineering effort but also a core business process (Berden, Brombacher et al. 2000). Increases in problem complexity result in higher demands with regard to costs and time associated with executing the simulation models required for evaluating system performance. As pointed out by Wheelwright and Clark (Wheelwright and Clark 1992), it is those firms that are able to develop and bring to market their products the fastest that are able to create a significant competitive advantage for themselves. Efforts aimed at reducing product development times, however, are faced with several challenges, identified by Lu (Lu 2002) as pertaining to (1) increases in product complexity, (2) increases in time-to-market (TTM) pressure, (3) globalization and segmentation, and (4) increasing customer demands. While a number of recent research activities focus on addressing the needs, underlying these challenges, a majority are aimed at meeting the intensive information requirements posed. One of the most notable recent efforts along these lines is that of PLM, which is taken to be a strategic business approach for the effective management and use of corporate intellectual capital (Edwards 2002; Fenves, Sriram et al. 2003; IBM 2004).

10.5.2 From Product LifeCycle to Process Lifecycle

In Figure 10-7, three key components of an enterprise's intellectual capital are presented including process information (top-left corner), product information (top-right corner) and the supporting PLM infrastructure (bottom) that consists of various software tools. Arrows between tools are used to represent flow of information among them. Dashed and solid lines are used to illustrate the fact that some of the links are more developed than others. As indicated in Figure 10-7, most of the elements of an engineering enterprise's intellectual capital relate to the acquisition of information pertaining to either product or process and the tools for transformation of this information. The infrastructure of PLM, as defined currently, centers on the integration of various software and associated hardware tools, ranging from CAD and analysis packages to PDM systems, etc., used for capturing and processing product information. To some extent, these tools are also employed for capturing information relating to the underlying design processes.

PLM efforts thus far have been focused on integration and interoperability. Although some of the relationships depicted by dashed and solid lines in Figure 10-7 have been implemented successfully, it is our belief that the effective management of a product's lifecycle (required for effective design process optimization) extends beyond ensuring the seamless flow of information between tools and requires a systems-based perspective of the entire engineering enterprise. Consequently we assert the importance of designing the design process alongside the product in PLM.

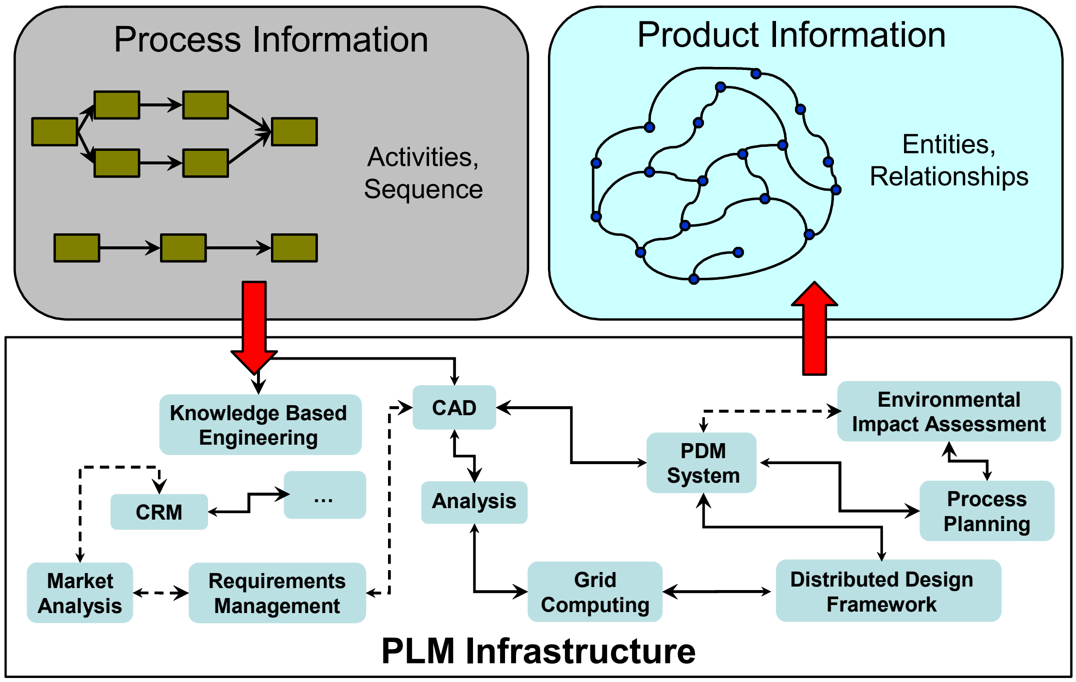


Figure 10-7 – Integrating sources of intellectual capital in an engineering enterprise (product information, process information, and PLM tools)

Although design processes play a crucial role in PLM, integrating the design of "design processes" with the product has received little attention. Additionally, while it is true that the potential of leveraging components of existing products towards developing new products has been exploited, the possibility of leveraging PLM sub-processes in new product realization scenarios is substantial. Thus, as an engineering enterprise becomes increasingly concerned with meeting the dynamic requirements of a global marketplace, closer attention must be paid to the mechanisms underlying product development. Perhaps the most crucial of these mechanisms is the design process. In terms of the engineering enterprise, this translates to the need for a systematic means of development for original, adaptive, variant, and derivative products. Although much attention has been paid to addressing this issue from a product-centric perspective by exploiting the reusability and scalability of products through product platform and product family design, not much attention has been paid to an engineering enterprise's primary resource commitment – ***the design process and its design***. In this dissertation we make an effort towards filling this research gap.

Many emerging approaches to PLM are concerned solely with lifecycle considerations as they relate to a single product. Considering that most engineering enterprises strive to maximize product portfolio diversity, a perspective of PLM focusing on the accommodation of the diverse and constantly changing needs of a global consumer base may be appropriate. Taking a step back, the question becomes: "*How can a company ensure the effective use of resources across the entirety of its product portfolio, especially as markets evolve with time?*" To be successful in such continuously changing marketplaces, it is essential to address not only current customer requirements, but also accommodate impending changes. With this in mind, ***it is emphasized that design processes should be viewed as constituting the strategy for developing a product***, given a set of requirements. Satisfying changing customer requirements is thus subject to one's ability to adapt the underlying design processes. This is true whether referring to a single original design or an adaptive, a variant, or a derivative design resulting there from, as supported by the assertion of Herbert Simon that "… design process strategies can affect not only the efficiency with which resources for designing are used, but also the nature of final design as well" (Simon 1996). The design of design processes thus constitutes a fundamental prerequisite for the strategic deployment of products and the effective consideration of their respective lifecycle considerations.

10.5.3 Closing Thoughts

The future basis for competition is likely to rest on an enterprise's ability to anticipate and quickly respond to market shifts and changes. This requires the effective leveraging of resources for optimum utilization of available assets both in terms of product information and the design processes. Considering that the bulk of the effort involved in product development lies in perfecting the underlying processes, they should be considered to be an enterprise's primary resource. Consequently, more attention must be paid to the manner in which these processes are designed and optimized.

Our premise, in this dissertation, is that *design processes* are an integral part of its intellectual capital. Accordingly, we establish the design of design processes (together with product design) as a critical factor in addressing lifecycle considerations of an evolving product portfolio. Five key requirements for enabling the design of design processes include identification of design process goals, process related decisions, information transformations, and computational models thereof, design process configuration, quantification of design process impact, and the integration of product and process-centric perspectives. Each of these is a potential avenue for concentrating the future efforts of the engineering optimization research community. Hence, attaining and retaining a competitive edge is likely to be a function of a company's agility in adapting existing design processes to the realization of adaptive, variant, derivative, and even original products. We note that the vision articulated in this section for the engineering design community is not meant to replace current efforts in the PLM arena. Instead, the

aim is to augment these efforts via the inclusion of design process related intellectual capital and emphasis on process improvement, thereby enhancing the overall agility of the engineering enterprise.

We assert that managing the lifecycle of a design process will have much greater leverage than merely considering the design of products in isolation. Hence, we believe that the vision and direction provided in this section are fundamental to the success of next generation agility in global enterprises. Considering the current scope PLM, this direction is extremely important. We thus envision extending the focus of PLM to include the lifecycle considerations of the design process, moving towards Design Process Lifecycle Management (DPLM). Considering the comprehensive nature of design processes, the underlying research problem is to manage and reuse *process knowledge* as a prime component of the intellectual capital. We must thus ask ourselves:

- To what extent can *families of processes* be modeled, captured, and reused?
- How can *top-down* design of engineering design processes be reconciled with *bottom-up* design of process components?
- How can all processes factoring into the value chain be *designed systematically* (e.g., engineering design processes, supply chain processes, etc.)?
- How can product information be reconciled with processes at various levels of abstraction in an *entire global enterprise*?

In closing, we leave you with the following thought -

"*Vision without action is merely a dream.*
Action without vision just passes the time.
Vision with action can change the world."
-- Joel A. Barker.

REFERENCES

Alexander, C., S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel, 1977, *A Pattern Language*, New York, Oxford University Press.

Allen, J. K., 1996, "The Decision to Introduce New Technology: The Fuzzy Preliminary Selection DSP", *Engineering Optimization*, 16(1), 61-77.

Asay, J. R. and M. Shahinpoor, 1993, *High-Pressure Shock Compression of Solids*, New York, Springer-Verlag.

Aughenbaugh, J. M., J. M. Ling and C. J. J. Paredis, 2005, "The Use of Imprecise Probabilities to Estimate the Value of Information in Design", *2005 ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, USA. Paper No: IMECE2005-81181.

Austin, R. A., 2005, "*Numerical Simulation of the Shock Compaction of Microscale Reactive Particle Systems*", MS Thesis, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta.

Badhrinath, K. and S. S. Rao, 1996, "Modeling for Concurrent Design Using Game Theory Formulations", *Concurrent Engineering: Research and Applications*, 4(4), 389-399.

Bahler, D., C. Dupont and J. Bowen, 1994, "Mediating Conflict in Concurrent Engineering with a Protocol Based on Utility, " *Concurrent Engineering: Research and Applications, Special Issue on Conflict Management in Concurrent Engineering*, 2, 197-207.

Ballew, W. D., 2004, "*Taylor Impact Test and Penetration of Reinforced Concrete Targets by Cylindrical Composite Rods*", MS Thesis, Engineering Science and Mechanics, Virginia Tech, Blacksburg, VA.

Balling, R. J. and J. S. Sobieski, 1996, "Optimization of Coupled Systems: A Critical Review of Approaches", *AIAA Journal*, 34(1), 6-17.

Bascaran, E., R. B. Bannerot and F. Mistree, 1989, "Hierarchical Selection Decision Support Problems in Conceptual Design", *Engineering Optimization*, 14, 207-238.

Baskaran, E., 1990, "*A Model for the Conceptual Design of Thermal Systems; Concurrent Decisions in Designing the Concept*", Ph.D., University of Houston, Houston, Texas.

Baskaran, E., R. B. Bannerot and F. Mistree, 1989, "Hierarchical Selection Decision Support Problems in Conceptual Design", *Engineering Optimization*, 14, 207-238.

Benson, D. J., 1995, "A Multi-Material Eulerian Formulation for the Efficient Solution of Impact and Penetration Problems", *Computational Mechanics*, 15, 558-571.

Berden, T. P. J., A. C. Brombacher and P. C. Sander, 2000, "The Building Bricks of Product Quality: An Overview of Some Basic Concepts and Principles", *International Journal of Production Economics*, 67, 3-15.

Bowen, J. and D. Bahler, 1992. Task Coordination in Concurrent Engineering. *Enterprise Integration Modeling*, MIT Press.

Bradley, S. R. and A. M. Agogino, 1994, "An Intelligent Real Time Design Methodology for Component Selection: An Approach to Managing Uncertainty", *Journal of Mechanical Design*, 116, 980-988.

Braha, D. and O. Maimon, 1997, "The Design Process: Properties, Paradigms, and Structure", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(2), 146-166.

Braha, D. and O. Maimon, 1998, *A Mathematical Theory of Design: Foundations, Algorithms, and Applications*, Boston, Kluwer Academic Publishers.

Braha, D. and Y. Reich, 2003, "Topological Structures for Modeling Engineering Design Process", *Research in Engineering Design*, 14(4), 185-199.

Bras, B., Mistree, F., 1991, "Designing Design Processes in Decision-Based Concurrent Engineering", *SAE Transactions Journal of Materials & Manufacturing*, 451-458.

Bras, B. A., 1992, "Designing Design Processes for Decision-Based Concurrent Engineering", *CERC's First Workshop on Product Development, Process Modeling and Characterization*, Morgantown, West-Virginia, Concurrent Engineering Research Center.

Bras, B. A., 1992, "*Foundations for Designing Decision-Based Design Processes*", Ph.D. Dissertation, University of Houston, Houston, Texas.

Bras, B. A. and F. Mistree, 1991. Designing Design Processes in Decision-Based Concurrent Engineering. *SAE Transactions, Journal of Materials & Manufacturing (SAE Paper 912209)*. Warrendale, Pennsylvania, SAE International. 100, 451-458.

Browning, T. R. and S. D. Eppinger, 2002, "Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development", *IEEE Transactions on Engineering Management*, 49(4), 428-442.

Buede, D. M., 2000, *The Engineering Design of Systems: Models and Methods*, New York, John Wiley & Sons, Inc.

Cantamessa, M. and A. Villa, 2000, "Product and Process Design Effort allocation in Concurrent Engineering", *International Journal of Production Research*, 38(14), 3131-3147.

Chandrasekaran, B., (1990), "Design Problem Solving: A Task Analysis", AI Magazine, Winter 1990**:** 59-71.

Chanron, V. and K. Lewis, 2003, "A Study of Convergence in Decentralized Design", *ASME Design Automation Conference*, Chicago, IL. Paper No: DETC03/DAC-48782.

Chanron, V. and K. Lewis, 2004, "Convergence and Stability of Distributed Design of Large Systems", *ASME Design Automation Conference*, Salt Lake City, UT. Paper No: DETC2004-57344.

Chanron, V., T. Singh and K. Lewis, 2004, "An Investigation of Equilibrium Stability in Decentralized Design Using Nonlinear Control Theory", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, USA. Paper No: AIAA-2004-4600.

Chen, W., 1995, "*A Robust Concept Exploration Method for Configuring Complex Systems*", PhD Dissertation, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Chen, W., J. K. Allen, D. Mavris and F. Mistree, 1996, "A Concept Exploration Method for Determining Robust Top-Level Specifications", *Engineering Optimization*, 26(2), 137-158.

Chen, W., J. K. Allen and F. Mistree, 1997, "A Robust Concept Exploration Method for Enhancing Productivity in Concurrent Systems Design", *Concurrent Engineering: Research and Applications*, 5(3), 203-217.

Chen, W., J. K. Allen, K. L. Tsui and F. Mistree, 1996, "A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors", *ASME Journal of Mechanical Design*, 118, 478-485.

Chen, W., R. Garimella and N. Michelena, 2001, "Robust Design for Improved Vehicle Handling Under a Range of Maneuver Conditions", *Enigneering Optimization*, 33(3), 2-22.

Chen, W. and K. Lewis, 1999, "A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design", *AIAA Journal*, 37(8), 982-989.

Choi, H.-J., 2005, "*A Robust Design Method for Model and Propagated Uncertainty*", PhD Dissertation, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta.

Choi, H.-J., R. Austin, J. K. Allen, D. L. McDowell and F. Mistree, 2004, "An Approach for Robust Micro-Scale Materials Design under Unparameterizable Variability", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY. Paper No: AIAA-2004-4331.

Cochran, J. K., K. J. Lee, D. L. McDowell and T. H. Sanders, 2000, "Low Density Monolithic Honeycombs by Thermal Chemical Processing", *Proceedings of the 4th Conference on Aerospace Materials, Processes, and Environmental Technology*, Huntsville, AL.

Coyne, R. D., (1990), "Design Reasoning: Without Explanations", AI Magazine, Winter 1990**:** 72-80.

Dolbow, J., M. A. Khaleel and J. Mitchell, 2004, "*Multiscale Mathematics Initiative: A Roadmap*", U.S. Department of Energy, PNNL-14966.

Eastman, C. M., A. H. Bond and S. C. Chase, 1991, "A Formal Approach for Product Model Information", *Research in Engineering Design*, 2(2), 65-80.

Edwards, J. D., 2002, "*Product Life Cycle Management: A White Paper*", *http://whitepapers.zdnet.co.uk/0,39025945,60040140p-39000604q,00.htm*.

Elmaghraby, S. E., 1995, "Activity Nets: A Guided Tour Through Some Recent Developments", *European Journal of Operational Research*, 82(3), 383-408.

Engenious Inc., 2004, "*FIPER*", *http://www.engineous.com/product_FIPER.htm*.

Engineous Inc., 2004, "*iSIGHT, Version 8.0*", *http://www.engineous.com/product_iSIGHT.htm*.

Eppinger, S., D. E. Whitney, R. P. Smith and D. A. Gebala, 1994, "A Model-based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, 6(1), 1-13.

Eppinger, S. D., 1991, "Model-based Approaches to Managing Concurent Engineering", *Journal of Engineering Design*, 2(4), 283-290.

Eppinger, S. D. and V. Salminen, 2001, "Patterns of Product Development Interactions", *International Conference on Engineering Design (ICED 01)*, Glasgow.

Eppinger, S. D., D. E. Whitney and R. P. Smith, 1990, "Organizing the Tasks in Complex Design Projects", *ASME Conference on Design Theory and Methodology*, Chicago, IL.

Evbuomwan, N. F. O., S. Sivaloganathan and A. Jebb, 1996, "A Survey of Design Philosophies, Models, Methods, and Systems", *Proceedings: Institution of Mechanical Engineers*, 210, 301-319.

Fenves, S. J., 2001, "*A Core Product Model for Representing Design Information*", NIST, Gaithersburg, NISTIR Report 6736.

Fenves, S. J., D. Sriram, R. Sudarsan and F. Wang, 2003, "A Product Information Modeling Framework for Product Lifecycle Management", *International Symposium on Product Lifecycle Management*, Bangalore, India.

Finger, S. and J. R. Dixon, 1989, "A Review of Research in Mechanical Engineering Design. Part 1: Descriptive, Prescriptive, and Computer-Based Models of Design Processes", *Research in Engineering Design*, 1, 51-67.

Finger, S. and J. R. Dixon, 1989, "A Review of Research in Mechanical Engineering Design. Part 2: Representations, Analysis, and Design for the Life Cycle", *Research in Engineering Design*, 1, 121-137.

Gamma, E., R. Helm, R. Johnson and J. Vlissides, 2000, *Design Patterns - Elements of Reusable Object Oriented Software*, Addison-Wesley Professional.

Gebala, D. A. and S. D. Eppinger, 1991, "Methods for Analyzing Design Procedures", *ASME Design Theory and Methodology*, Miami, FL. Paper No: DE-Vol 31.

Gero, J. S., (1990), "Design Prototypes: A Knowledge Representation Schema for Design", AI Magazine, Winter 1990**:** 26-36.

Gorti, S. R., A. Gupta, G. J. Kim, R. D. Sriram and A. Wong, 1998, "An Object-Oriented Representation for Product and Design Process", *Computer Aided Design*, 30(7), 489-501.

Hacker, K. and K. Lewis, 1998, "Using Robust Design Techniques to Model the Effects of Multiple Decision-Makers in a Design Process", *1998 ASME Design Engineering Technical Conferences, Advances in Design Automation Conference*, Atlanta, GA. Paper No: DETC98/DAC-5604.

Hansen, E. and G. W. Walster, 2004, *Global Optimization Using Interval Analysis*, MIT Press, Cambridge.

Hasan, O. A. and M. C. Boyce, 1995, "A Constitutive Model for the Nonlinear Viscoelastic Viscoplastic Behavior of Glassy Polymers", *Polymer Engineering and Science*, 35(4), 331-344.

Hayek, F. A., 1945, "The Use of Knowledge in Society", *American Economic Review*, 35(4), 519-530.

Hayes, A. M., A. Wang, B. M. Dempsey and D. L. McDowell, 2001, "Mechanics of Linear Cellular Alloys", *Proceedings of IMECE, International Mechanical Engineering Congress and Exposition*, New York, NY.

Hazelrigg, G. A., 1998, "A Framework for Decision-Based Engineering Design", *Journal of Mechanical Design*, 120(4), 653-658.

Hernández, G., 1998, "*A Probabilistic-Based Design Approach with Game Theoretical Representations of the Enterprise Design Process*", MS Thesis, Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Hernández, G., C. C. Seepersad, J. Allen and F. Mistree, 2002, "A Method for Interactive Decision-Making in Collaborative, Distributed Engineering Design", *International Journal of Agile Manufacturing Systems*, 5(2), 47-65.

Hernández, G., C. C. Seepersad, J. K. Allen and F. Mistree, 2002, "Framework for Interactive Decision-Making in Collaborative, Distributed Engineering Design", *International Journal of Advanced Manufacturing Systems (IJAMS) Special Issue on Decision Engineering*, 5(2), 47-65.

Herrmann, J. W. and L. C. Schmidt, 2002, "Viewing Product Development as a Decision Production System", *ASME Design Theory and Methodology Conference*, Montreal, Canada. Paper No: DETC2002/DTM-34030.

Houghton, J., L. M. Filho, D. Griggs and K. Maskell, 1997, "*An Introduction to Simple Climate Models used in the IPCC Second Assessment Report - IPCC Technical Paper II*", Inter-governmental Panel on Climate Change, http://www.ipcc.ch/pub/IPCCTP.II(E).pdf.

Howard, R., 1966, "Information Value Theory", *IEEE Transactions on Systems Science and Cybernetics*, SSC-2(1), 779-783.

Hsu, Y.-L., C.-Y. Tai and Y.-C. Chen, 2000, "A Design Process Model Based on Product States", *Journal of Chinese Society of Mechanical Engineers*, 21(4), 369-377.

IBM, 2004, "*IBM Solutions*", *http://www-1.ibm.com/businesscenter/us/solutions/solutionarea.jsp?id=9351*.

Integrated Definition for Functional Modeling, 1993, "*(IDEF 0)*", Federal Information Processing Standards Publication 183, *http://www.idef.com/Downloads/pdf/idef0.pdf*.

Kalsi, M., K. Hacker and K. Lewis, 1999, "A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems", *ASME Advances in Design Automation Conference*, Las Vegas, Nevada. Paper No: DETC99/DAC-8589.

Kamal, S. Z., 1990, "*The Development of Heuristic Decision Support Problems for Adaptive Design*", Ph.D. Dissertation, Department of Mechanical Engineering, University of Houston, Houston, Texas.

Kamal, S. Z., H. M. Karandikar, F. Mistree and D. Muster, 1987. Knowledge Representation for Discipline-Independent Decision Making. *Expert Systems in*

Computer-Aided Design. J. Gero. Amsterdam, Elsevier Science Publishers B.V., 289-321.

Karandikar, H. and F. Mistree, 1993. Modeling Concurrency in the Design of Composite Structures. *Structural Optimization: Status and Promise*. M. P. Kamat. Washington, D.C., AIAA, 769-806.

Keeney, R. L. and H. Raiffa, 1976, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, New York, John Wiley and Sons.

Kinslow, R., 1970, *High-Velocity Impact Phenomena*, New York, Academic Press.

Klepaczko, J. R., T. Sasaki and T. Kurokawa, 1993, "On Rate Sensitivity of Polycrystalline Aluminum at High Strain Rates", *Trans. Japan Soc. Aero. Space Sci.*, 36(113), 170-187.

Koch, P. N., 1997, "*Hierarchical Modeling and Robust Synthesis for the Preliminary Design of Large Scale Complex Systems*", PhD Dissertation, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Kron, G., 1963, *'Diakoptics', Piecewise Solution of Large Scale System of Equations*, London, MacDonald & Co.

Kusiak, A., N. Larson and J. Wang, 1994, "Reengineering of Design and Manufacturing Processes", *Computers and Industrial Engineering*, 26(3), 521-536.

Kusiak, A. and K. Park, 1990, "Concurrent Engineering: Decomposition and Scheduling of Design Activities", *International Journal of Production Research*, 28(10), 1883-1900.

Kusiak, A., J. Wang, D. He and C. Feng, 1995, "A Structured Approach for Analysis of Design Processes", *IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part A*, 18(3), 664-673.

Kusiak, A., J. Wang and D. W. He, 1996, "Negotiation in Constraint-Based Design", *Journal of Mechanical Design*, 118(4), 470-477.

Lawrence, D. B., 1999, *The Economic Value of Information*, New York, Springer.

Lee, H. and S. Whang, 1999, "Decentralized Multi-Echelon Supply Chains: Incentives and Information", *Management Science*, 45(5), 633-639.

Lewis, K. and F. Mistree, 1997, "Modeling Interaction in Multidisciplinary Design: A Game Theoretic Approach", *AIAA Journal*, 35(8), 1387-1392.

Liker, J., D. Sobek, A. Ward and J. Cristiano, 1996, "Involving Suppliers in Product Development in the US and Japan: Evidence for Set-Based Concurrent Engineering", *IEEE Transactions on Engineering Management*, 43(2), 165-178.

Lu, X., V. Narayanan and S. Hanagud, 2003, "Shock Induced Chemical Reactions in Energetic Structural Materials", *13th American Physical Society Topical Conference on Shock Compression of Condensed Matter*, Portland, Oregon.

Lu, Y., 2002, "*Analyzing Reliability Problems in Concurrent Fast Product Development Processes*", PhD Dissertation, Mechanical Engineering, Technische Unversiteit Eindhoven, Eindhoven, Netherlands.

Lyons, K. W., M. R. Duffey and R. C. Anderson, 1995, "*Product Development Process Modeling: A Study of Requirements, Methods and Research Issues*", National Institute of Standards and Technology, Report No: NISTIR 5745.

Magrab, E., 1997, *The Integrated Product and Process Design and Development*, Boca Raton, FL, CRC Press.

Maher, M. L., (1990), "Process Models for Design Synthesis", AI Magazine, Winter 1990**:** 49-58.

Maimon, O. and D. Braha, 1996, "On the Complexity of the Design Synthesis Problem", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 26(1), 142-151.

Marston, M., 2000, "*Game Based Design: A Game Theory Based Approach to Engineering Design*", PhD Dissertation, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Marston, M., J. K. Allen and F. Mistree, 2000, "The Decision Support Problem Technique: Integrating Descriptive and Normative Approaches in Decision Based Design", *Engineering Valuation and Cost Analysis*, 3, 107-129.

Marston, M. and F. Mistree, 2000, "Game-Based Design: A Game Theoretic Extension to Decision-Based Design", *ASME Design Theory and Methodology Conference*, Baltimore, MD. Paper No: DETC2000/DTM-14578.

Martin, M. V. and K. Ishii, 1997, "Design for Variety: Development of Complexity Indices and Design Charts", *ASME Design for Manufacturing Conference*, Sacramento, CA. Paper No: DETC97/DFM-4359.

Martin, M. V. and K. Ishii, 2000, "Design for Variety: A Methodology for Developing Product Platform Architectures", *ASME Design for Manufacturing Conference*, Baltimore, MD. Paper No: DETC2000/DFM-14021.

Merzhanov, A. G., 1966, "On Critical Conditions for Thermal Explosion of a Hot Spot", *Combustion and Flame*, 10, 341-348.

Meso Inc., 2005, "*Meso Inc.*" http://www.meso.com/.

MetOffice, 2005, "*Climate models*", *http://www.metoffice.com/research/hadleycentre/models/modeltypes.html*.

Miller, R. E., 2003, "Direct Coupling of Atomistic and Continuum Mechanics in Computational Materials Science", *International Journal of Computational Materials Science*, 1(1), 57-72.

Mistree, F., B. A. Bras, W. F. Smith and J. K. Allen, 1996, "Modeling Design Processes: A Conceptual, Decision-Based Perspective", *International Journal of Engineering Design and Automation*, 1(4), 209-221.

Mistree, F., O. F. Hughes and B. A. Bras, 1993. The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm. *Structural Optimization: Status and Promise*. M. P. Kamat. Washington, D.C., AIAA, 247-286.

Mistree, F., K. Lewis and L. Stonis, 1994, "Selection in the Conceptual Design of Aircraft", *5th AIAA/USAF/NASA/ISSMO Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, Panama City, FL, USA, AIAA. Paper No: pp. 1153-1166.

Mistree, F., D. Muster, J. A. Shupe and J. K. Allen, 1989, "A Decision-Based Perspective for the Design of Methods for Systems Design", *Recent Experiences in Multidisciplinary Analysis and Optimization*, Hampton, Virginia. Paper No: NASA CP 3031.

Mistree, F., D. Muster, J. A. Shupe and J. K. Allen, 1989, "A Decision-Based Perspective for the Design of Methods for Systems Design", *Recent Experiences in Multidisciplinary Analysis and Optimization*, Hampton, Virginia. Paper No: NASA CP 3031.

Mistree, F., W. F. Smith, B. Bras, J. K. Allen and D. Muster, 1990. Decision-Based Design: A Contemporary Paradigm for Ship Design. *Transactions, Society of Naval Architects and Marine Engineers*. Jersey City, New Jersey. 98, 565-597.

Mistree, F., W. F. Smith and B. A. Bras, 1993. A Decision-Based Approach to Concurrent Engineering. *Concurrent Engineering: Contemporary Issues and Modern Design Tools*. H. R. Paresai and W. G. Sullivan. New York, N.Y., Chapman & Hall, 127-158.

Mistree, F., W. F. Smith and B. A. Bras, 1993. A Decision-Based Approach to Concurrent Engineering. *Handbook of Concurrent Engineering*. H. R. Paresai and W. Sullivan. New York, Chapman & Hall, 127-158.

Mistree, F., W. F. Smith, B. A. Bras, J. K. Allen and D. Muster, 1990, "Decision-Based Design: A Contemporary Paradigm in Ship Design", *Transactions, Society of Naval Architects and Marine Engineers*, 98, 565-597.

Mistree, F., W. F. Smith, S. Z. Kamal and B. A. Bras, 1991, "Designing Decisions: Axioms, Models and Marine Applications", *Fourth International Marine Systems Design Conference*, Kobe, Japan, Society of Naval Architects of Japan.

Murnaghan, F. D., 1937, "Finite deformations of an elastic solid", *American Journal of Mathematics*, 49, 235-260.

Muster, D. and F. Mistree, 1988, "The Decision Support Problem Technique in Engineering Design", *International Journal of Applied Engineering Education*, 4(1), 23-33.

Muster, D. and F. Mistree, 1988, "The Decision Support Problem Technique in Engineering Design", *The International Journal of Applied Engineering Education*, 4(1), 22-33.

Myerson, R. B., 1991, *Game Theory: Analysis of Conflict*, Cambridge, MA, Harvard University Press.

Nell, J., 2003, "*STEP on a Page (ISO 10303)*", http://www.nist.gov/sc5/soap/.

Newcomb, P. J., B. A. Bras and D. W. Rosen, 1996, "Implications of Modularity on Product Design for the Life Cycle", *ASME Design Theory and Methodology Conference*, Irvine, California. Paper No: DETC-96/DTM-1516.

Pahl, G. and W. Beitz, 1996, *Engineering Design: A Systematic Approach*, New York, Springer-Verlag.

Panchal, J. H., M. G. Fernández, C. J. J. Paredis, J. K. Allen and F. Mistree, 2004, "Designing Design Processes in Product Lifecycle Management: Research Issues and Strategies", *ASME 2004 Computers And Information In Engineering Conference*, Salt Lake City, Utah. Paper No: DETC2004/CIE-57742.

Park, H. and M. R. Cutkosky, 1999, "Framework for Modeling Dependencies in Collaborative Engineering Process", *Research in Engineering Design*, 11(2), 84-102.

Patel, N., 2004, "*Intermediate Strain Rate Behavior of Two Structural Energetic Materials*", MS Thesis, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta.

Peak, R., J. Lubell, V. Srinivasan and S. Waterbury, 2004, "STEP, XML, and UML: Complementary Technologies", *Journal of Computing and Information Science in Engineering: Special Issue on Engineering Information Management to Product Lifecycle Management*, 4(4), 379-390.

Pedersen, K., J. Emblemsvag, R. Bailey, J. Allen and F. Mistree, 2000, "The 'Validation Square' - Validating Design Methods", *ASME Design Theory and Methodology Conference*, New York. Paper No: ASME DETC2000/DTM-14579.

Pennell, J. and M. Slusarczuk, 1989, "*An Annotated Reading List for Concurrent Engineering*", Institute for Defence Analysis, Alexandria, VA, Technical Report HQ 89-034130.

Phoenix Integration Inc., 2004, "*ModelCenter®, Version 5.0*", http://www.phoenix-int.com/products/ModelCenter.html.

Pimmler, T. U. and S. D. Eppinger, 1994, "Integration Analysis of Product Decompositions", *ASME Design Theory and Methodology*, Minneapolis, MN.

Poh, K. L. and E. Horvitz, 1993, "Reasoning about the Value of Decision Model Refinement: Methods and Application." *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence*, Washington DC, Morgan Kaufmann: San Francisco.

Prasad, B., 1996, *Concurrent Engineering Fundamentals: Integrated Product and Process Organization*, Upper Saddle River, NJ, Prentice Hall PTR.

Rao, S. S., 1987, "Game Theory Approach for Multiobjective Structural Optimization", *Computers and Structures*, 25(1), 119-127.

Rao, S. S. and T. I. Freihet, 1991, "A Modified Game Theory Approach to Multiobjective Optimization", *Journal of Mechanical Design*, 113(3), 286-291.

Rechtin, E. and M. W. Maier, 1997, *The Art of Systems Architecting*, Boca Raton: CRC Press.

Reddy, R. and F. Mistree, 1992. Modeling Uncertainty in Selection using Exact Interval Arithmetic. *Design Theory and Methodology 92*. L. A. Stauffer and D. L. Taylor eds. New York. DE-Vol. 42, 193-201.

Rudd, R. E. and J. Q. Broughton, 2000, "Concurrent Coupling of Length Scales in Solid State Systems", *Phys. Stat. Sol. (b)*, 217, 251-291.

Salingaros, N. A., 2000, "The Structure of Pattern Languages", *Architectural Research Quarterly*, 4, 149-161.

Sambu, S. P., 2001, "*A Design for Manufacture Method for Rapid Prototyping and Rapid Tooling*", M.S. Thesis, The GW Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia.

SBES Workshop Report, 2004, "*Simulation Based Engineering Science*", National Science Foundation, Arlington, VA.

Schlenoff, C., A. Knutilla and S. Ray, 1996, "*Unified Process Specification Language: Requirements for Modeling Process*", National Institute of Standards and Technology, Gaithersburg, MD, NISTIR 5910.

Scott, M. J., 1999, "*Formalizing Negotiation in Engineering Design*", PhD Dissertation, Mechanical Engineering, California Institute of Technology, Pasadena, CA, USA.

Scott, M. J. and E. K. Antonsson, 1996, "Formalisms for Negotiation in Engineering Design", *ASME Design Theory and Methodology Conference*, Irvine, CA. Paper No: DETC-96/DTM-1525.

Seepersad, C. C., 2001, "*A Utility-Based Compromise Decision Support Problem with Applications in Product Platform Design*", M.S. Thesis, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Seepersad, C. C., B. M. Dempsey, J. K. Allen, F. Mistree and D. L. McDowell, 2002, "Design of Multifunctional Honeycomb Materials", *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA. Paper No: Paper Number: AIAA-2002-5626.

Seepersad, C. C., K. Pedersen, J. Emblemsvag, R. R. Bailey, J. K. Allen and F. Mistree, 2005. The Validation Square: How Does One Verify and Validate a Design Method? *Decision-Based Design: Making Effective Decisions in Product and Systems Design*. W. Chen, K. Lewis and L. Schmidt. New York, NY.

Shimomura, Y., M. Yoshioka, H. Takeda, Y. Umeda and T. Tomiyama, 1998, "Representation of Design Object Based on the Functional Evolution Process Model", *Journal of Mechanical Design*, 120(2), 221-229.

Simon, H. A., 1996, *The Sciences of the Artificial*, Cambridge, Mass., MIT Press.

Simpson, T., Lautenschlager, U., Mistree, F., 1998. Mass Customization in the Age of Information: The Case for Open Engineering Systems. *The Information Revolution Current and Future Consequences*. W. H. R. Alan L. Porter. Greenwhich Connecticut, Ablex Publishing Corporation, 49-74.

Simpson, T. W., W. Chen, J. K. Allen and F. Mistree, 1996, "Conceptual Design of a Family of Products Trough the Use of the Robust Concept Exploration Method", *AIAA/USAF/NASA/ISSMO Symposium on Multidsicplinary Analysis and Optimization*, Bellevue, WA.

Simpson, T. W., J. R. A. Maier and F. Mistree, 2001, "Product Platform Design: Method and Aplication", *Research in Engineering Design*, 13(1), 2-22.

Simpson, T. W., D. Rosen, J. K. Allen and F. Mistree, 1998, "Metrics for Assessing Design Freedom and Information Certainty in the Early Stages of Design", *Journal of Mechanical Design*, 120(4), 628-635.

Smith, R. P. and J. A. Morrow, 1999, "Product Development Process Modeling", *Design Studies*, 20(3), 237-261.

Sobek, D. K. and A. C. Ward, 1996, "Principles from Toyota's Set-Based Concurrent Engineering Process", *ASME Design Theory and Methodology Conference*, Irvine, CA. Paper No: DETC-96/DTM-1510.

Steward, D. V., 1981, "The Design Structure System: A Method for Managing the Design of Complex Systems", *IEEE Transactions on Engineering Management*, EM 78(3), 71-74.

Struble, C. L., E. Bascaran, R. B. Bannerot and F. Mistree, 1989, "Compromise: A Multiobjective Hierarchical Approach to the Design of Spacecraft Thermal Control Systems", *ASME Computers in Engineering Conference*, Anaheim, CA.

Suh, N. P., 1990, *Principles of Design*, Oxford, U.K., Oxford University Press.

Taguchi, G., 1986, *Introduction to Quality Engineering*, New York, UNIPUB.

Takeda, H., P. Veerkamp, T. Tomiyama and H. Yoshikawa, (1990), "Modeling Design Processes", AI Magazine, Winter 1990**:** 37-48.

Takle, E. S. and D. T. Kao, 1998, "A Synthesis of Models for Describing Multi-scale Interactions within Natural Systems", *MISSION EARTH '98: Modeling and Simulation of the Earth System*, San Diego, California, USA.

Taylor, G. I., 1946, "The Testing of Materials at High Rates of Loading", *J. Inst. Civil Eng.*, 26, 486-519.

Taylor, G. I., 1948, "The Use of Flat Ended Projectiles for Determining Yield Stress. I: Theoretical Consideration", *Proc. R. Soc. Lond.*, A194, 289-299.

Thompson, G. L., 1953, "Signaling strategies in n-person games", *Annals of Mathematics Studies*, 28.

Thurston, D. L., 1999, "Real and Perceived Limitations to Decision Based Design", *ASME Design Theory and Methodology*, Las Vegas, NV, USA, ASME. Paper No: DETC99/DTM-8750.

Tomiyama, T. and H. Yoshikawa, 1986, "Extended General Design Theory", *Design Theory for CAD, Proceedings of the IFIP WG5.2 Working Conference 1985*, Elsevier, North-Holland, Amsterdam.

Ullman, D. G., 1992, "A Taxonomy for Mechanical Design", *Research in Engineering Design*, 3(3), 179-189.

Umeda, Y., H. Takeda, T. Tomiyama and H. Yoshikawa, 1990. Function, Behavior, and Structure. *AIENG '90 Applications of AI in Engineering*, Computational Mechanics Publications and Springer Verlag, 177-193.

Vanderbei, R. J., 1996, *Linear Programming - Foundations and Extensions*, Springer, 0792373421.

Von Neumann, J. and O. Morgenstern, 1947, *The Theory of Games and Economic Behavior*, Princeton, NJ, Princeton University Press.

Ward, A., J. Liker, J. Cristiano and D. Sobek, 1995, "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster", *Sloan Management Review*, 36(3), 43-61.

Warfield, J. N., 1973, "Binary Matrices in System Modeling", *IEEE Transactions on Systems, Man, and Cybernetics*, 3, 441-449.

Watson, R. T., D. L. Albritton, T. Barker, I. A. Bashmakov, O. Canziani, R. Christ, U. Cubasch, O. Davidson, H. Gitay and Others, 2001, "*Climate Change 2001: Synthesis Report*", http://www.ipcc.ch/pub/un/syreng/spm.pdf.

Weinan, E. and B. Engquist, 2003, "Multiscale Modeling and Computation", *Notices of the AMS*, 50(9), 1062-1070.

Weinan, E., B. Engquist and Z. Huang, 2003, "Heterogeneous Multiscale Method: A General Methodology for Multiscale Modeling", *Physical Review B*, 67(092101), 1-4.

Wheelwright, S. C. and K. B. Clark, 1992, *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*, New York, Free Press.

Whitney, D. E., 1996, "Why Mechanical Design Cannot be Like VLSI Design", *Research in Engineering Design*, 8(3), 125-138.

Winner, R. I., J. P. Pennell, H. E. Bertrand and M. M. G. Slusarczuk, 1988, "*The Role of Concurrent Engineering in Weapons System Acquisition*", Institute for Defense Analysis, Alexandria, VA, IDA Report R-338.

Wood, W. H., 2000, "Quantifying Design Freedom in Decision Based Conceptual Design", *ASME Design Theory and Methodology Conference*, Baltimore, Maryland. Paper No: DETC2000/DTM-14577.

Wood, W. H., 2001, "A View of Design Theory and Methodology from the Standpoint of Design Freedom", *ASME Design Theory and Methodology Conference*, Pittsburgh, Pennsylvania. Paper No: DETC2001/DTM-21717.

Xiao, A., S. Zeng, J. K. Allen, D. W. Rosen and F. Mistree, 2002, "Collaborating Multi-Disciplinary Decision-Making using Game Theory and Design Capability Indices", *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA. Paper No: AIAA-2002-5617.

Zeng, Y. and P. Gu, 1999, "A Science Based Approach to Product Design Theory Part 1: Formulation and formalization of Design Process", *Robotics and Computer Integrated Manufacturing*, 15(6), 331-339.

Zeng, Y. and P. Gu, 1999, "A Science Based Approach to Product Design Theory Part II: Formulation of Design Requirements and Products", *Robotics and Computer Integrated Manufacturing*, 15(6), 341-352.

VITA

Jitesh H Panchal was born in Godhra, India in 1979. He grew up in Faridabad, India and received his high school education in the same city until 1996. He attended the Indian Institute of Technology at Guwahati, India from 1996-2000 and earned his Bachelor of Technology degree in Mechanical Engineering. After his undergraduate education, he worked as a software engineer at Interra Information Technologies, Noida, India from 2000-2001. He joined the graduate program in Mechanical Engineering at Georgia Institute of Technology, Atlanta, USA in 2001. After receiving his Master of Science degree in Mechanical Engineering at Georgia Institute of Technology in 2003, he continued his graduate studies in the field of Engineering Design. He received his PhD in Mechanical Engineering in 2005. After his PhD, Jitesh is interested in an academic position at a university that emphasizes excellence in both research and teaching.